



**Universidad de las Ciencias Informáticas
FACULTAD 10**

Propuesta de Expediente de Proyecto para los proyectos del Área Temática Herramientas para la Teleformación que utilizan programación estructurada.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autores:

Lilian Vigoa Machín

Yorleni Ramos Pérez

Tutores:

Ing. Yolanda Sardiñas Suárez

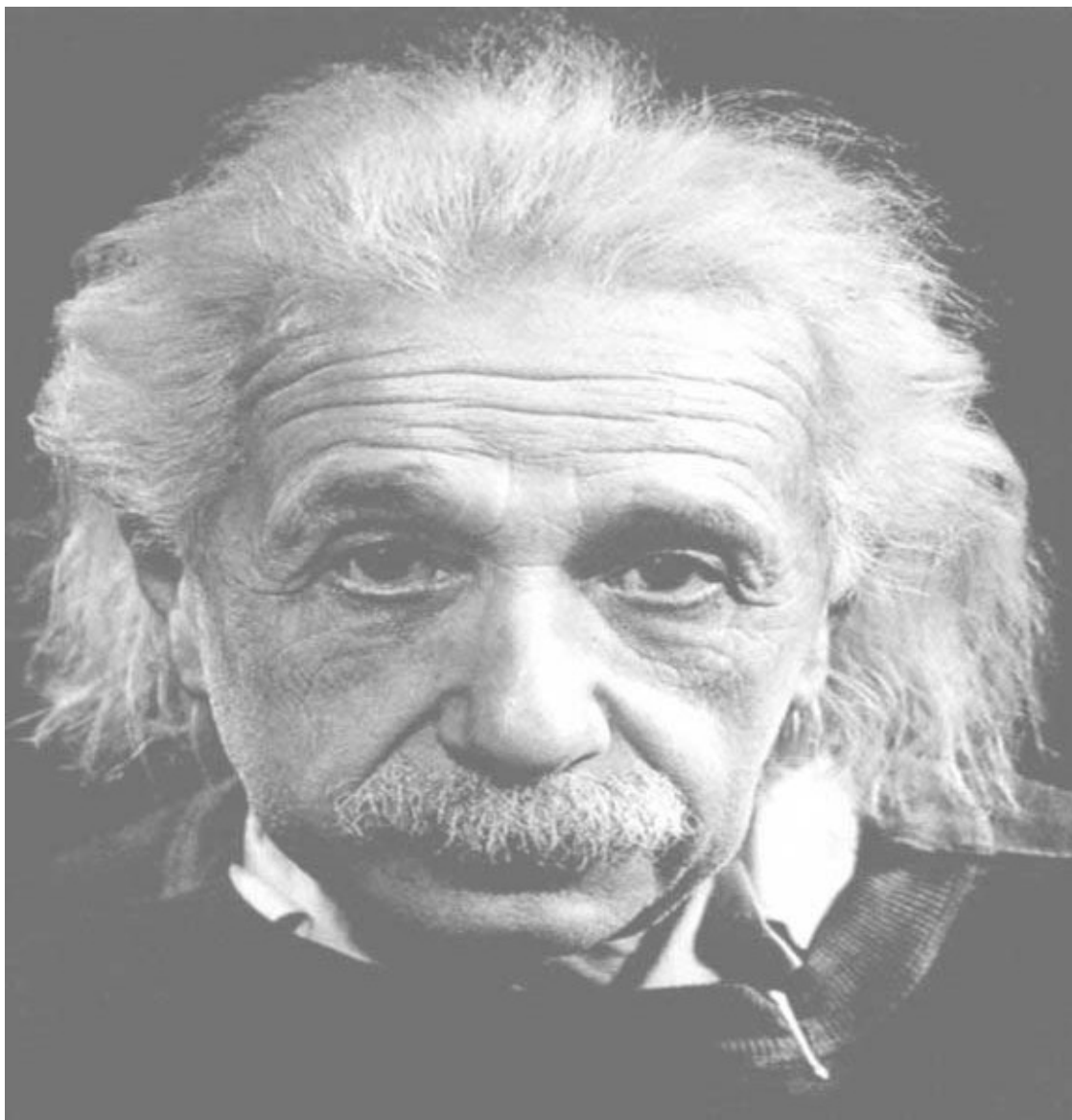
Ing. Jesús Hidalgo Guillén

Co-Tutor:

Ing. Susel Cañete Pollán

Ciudad de La Habana

Junio 2009



“No hay cosas sin interés, tan sólo personas incapaces de interesarse.”

Albert Einstein

Ing. Yolanda Sardiñas Suárez: Profesora de Contabilidad y Finanzas. Facultad 10, Universidad de Ciencias Informáticas. Carretera a San Antonio de los Baños Km. 2 ½, Torrens, Boyeros, Ciudad de La Habana. Cuba. Teléfono: (53) (07) 835 8282. yssuarez@uci.cu Graduada como Ingeniera en Ciencias Informáticas, Universidad de las Ciencias Informáticas, 2008. Culminó sus estudios con índice académico de 4.68 puntos. Se encontraba vinculada a proyectos productivos desde su tercer año de carrera, ocupando el cargo de analista en los distintos grupos en los que trabajó, pertenecientes al Área Temática Herramientas para la Teleformación. Una vez graduada continuó en dicha universidad ocupando el cargo de Analista Principal del Grupo “Personalización y Extensión de Herramientas para la Teleformación”, perteneciente al Área Temática Herramientas para la Teleformación y vinculada a la docencia. Categoría Docente Adiestrada.

Ing. Jesus Hidalgo Guillén: Profesor de la asignatura de Introducción a la Programación. Facultad # 10, Universidad de las Ciencias Informática (UCI). Ciudad de La Habana. Cuba. jhidalgo@uci.cu Ingeniero en Ciencias Informáticas, UCI, 2007. Culminó sus estudios con un índice académico de 4.5 puntos. Fue Alumno Ayudante en la Universidad de las Ciencias Informática en las Disciplinas de Programación. Se encontraba vinculado a proyectos productivos desde su tercer año de carrera, ocupando cargos de desarrollador, jefe de proyecto y líder del grupo “Personalización de Herramientas para la Teleformación”, perteneciente al Área Temática Herramientas para la Teleformación. Una vez graduado continuó en dicha universidad ocupando el cargo de líder del grupo “Personalización de Herramientas para la Teleformación” y vinculada a la docencia.

Ing. Susel Cañete Pollán: Profesora de Práctica Profesional 5 en la Facultad 10. Graduada como Ingeniera en Ciencias Informáticas en La Universidad de Ciencias Informáticas (UCI). Carretera a San Antonio de los Baños Km. 2 ½, Torrens, Boyeros, Ciudad de La Habana. Cuba. Teléfono: (53) (07) 835 8282. scanete@uci.cu. Categoría Docente Adiestrada, sin Categoría Científica. Es profesora de La Universidad de Ciencias Informáticas (UCI) desde su graduación en el año 2007. Su trabajo de diploma fue desarrollado en la línea investigativa de las técnicas formales, y ha sido tutora de otras tesis relacionadas con el tema. Trabajó desde el pasado curso y hasta principio del presente como miembro del grupo de calidad en el Área Temática Herramientas para la Teleformación; actualmente se encuentra desempeñando el rol de aseguradora de calidad en el proyecto Calidad.

Agradezco a mi madre Ana Caridad Machín por brindarme la oportunidad de demostrarle que todo su valor y sacrificio durante 23 años no fue en vano, por confiar en mí ciegamente, por ser la parte más importante de mi corazón, por cuidar de mí, por haber sido madre y padre durante toda mi vida y por enseñarme que la amistad y el amor se reúnen en una sola palabra, MADRE.

Agradezco a mis hermanas Ana Iris, Ana Yilian y Zuzel por ayudarme siempre, por ser incondicionales, por permanecer a mi lado en momentos difíciles, y por enseñarme que las mejores hermanas no son las que llevan la misma sangre, sino las que escoge el alma.

Le agradezco a Luis Armando por llegar a mi vida en ese momento que me encontraba ciega de locura, por cuidar siempre de mí, por amarme, por convertirse en la luz de mis ojos, por enseñarme que el amor verdadero si existe, y por permitirme pasar el resto de mi vida a su lado. Te Amo.

Agradezco a mis abuelos, Hilda Elisa y Guillermo por quererme tanto, por demostrarme siempre su cariño, y por dejarme ocupar un lugar en sus corazoncitos.

Agradezco a Fidel Núñez por hacer tantas cosas por mi familia, por cuidar a mi mamá, y por ayudarme en la toma de todas mis decisiones.

Agradezco a Lidia Medina por ser mi segunda mamá, por enfrentar todo a mi lado, por cuidarme cuando estaba enferma, por ser hermana, madre y amiga.

Agradezco a Nelly, la mujer que se merece este resultado, que se merece mi título, le agradezco toda su dedicación y todo el bien que hace.

Agradezco a mi tutora Yolanda Sardiñas por todas sus horas de dedicación y esmero, por sus regaños, por su paciencia. A mi tutor Jesús Hidalgo por no dejar que me rindiera, por permanecer a nuestro lado siempre.

Agradecimientos

Agradezco a Roxana Cañizares, por ser quien es, por brindarme siempre su tiempo, por ser amiga y compañera.

Agradezco al profesor Gustavo Javier por inculcarme valores, a Daynel Mármol por brindarme esa ayuda incondicional, a Daymí Tamayo, por ser ejemplo a seguir, a Marilín por acompañarme en mis lágrimas, a Héctor Delgado por ser el gran amigo que es, Le agradezco a todos mis compañeros de aula, por siempre estar en el momento preciso, a todas mis amistades por siempre darme lo mejor de sí.

Le agradezco al Comandante en Jefe Fidel Castro, por darme la oportunidad de estudiar en esta escuela que significa toda una vida para mí, y por enseñarme a ser lo que soy hoy, una Mujer de Bien.

A todos Muchas Gracias

Lily.

Agradezco a mis padres María Pérez y Martín Ramos porque sin ellos este sueño no se hubiese hecho realidad, gracias por darme tanto amor, por estar presentes en cada momento de mi vida, por ser los mejores padres del mundo.

Agradezco a mis tutores, Yolanda Sardiñas Suarez, Jesús Hidalgo y Susel Cañete Pollán por apoyarnos, guiarnos y confiar en nosotras durante nuestro largo camino de estudiante, por ser incondicionales.

Agradezco a mi hermana por apoyarme y guiarme en la vida, a mi familia, por todo el amor que me han brindado en todos estos años, agradezco a mi novio por mantener siempre mi pensamiento, por su comprensión y dedicación.

Agradezco a todos mis amigos y compañeros que me han acompañado a lo largo de estos 5 años de mi carrera, por siempre estar a mi lado y ayudarme cuando lo he necesitado, por ser como una familia para mí, siempre estarán conmigo en mi corazón.

A todos Muchas Gracias

Yorleni.

A mi madre, que es mi razón de ser.

A mis abuelitos, que son la Luz de mi Vida.

A Luis Armando, por convertirse en el Amor de mi Vida.

A Dios, por ser tan grande y milagroso.

Lily.

A mi madre, por ser mi razón de ser.

*A mi padre, por darme la confianza en mí misma para
siempre seguir adelante.*

A mi hermana Yohany, por estar siempre presente.

Yorleni.

Las metodologías de desarrollo de software constituyen el marco de trabajo de la ingeniería de software usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Además, se encargan de elaborar estrategias de desarrollo de software que promuevan prácticas adoptivas, centradas en las personas o los equipos.

En este trabajo se realiza una propuesta de Expediente de Proyecto basado en la metodología de desarrollo de software Métrica V3, para agrupar y organizar todos los artefactos que se generan durante el proceso de desarrollo de software de los proyectos que utilizan programación estructurada en el Área Temática Herramientas para la Teleformación de la Universidad de las Ciencias Informáticas (UCI). Para la elaboración de este expediente se tuvo en cuenta los artefactos que se generan en esta metodología, los Lineamientos de Calidad de Software establecidos en la UCI, así como el Expediente de Proyecto definido por la misma.

Además, se aplicó la propuesta de Expediente de Proyecto de Instalación en un proyecto perteneciente al Grupo de Personalización y Extensión de Plataformas para la Gestión del Aprendizaje, se analizaron los resultados alcanzados a través de las encuestas realizadas a diferentes integrantes del proyecto de Plataformas para la Gestión del Aprendizaje y se obtuvieron resultados de las encuestas realizadas.

Palabras clave: Expediente de Proyecto, Metodología, Lineamientos, Métrica V3.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPITULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
Introducción	5
1. Situación actual de las metodologías	5
1.1 Metodologías Estructuradas	7
1.1.1 Historia.....	7
1.1.1.1 Clasificación	9
1.1.1.2 Características	9
1.1.1.3 Ejemplos de Metodologías Estructuradas.....	10
1.2 Lineamientos de Calidad de Software propuesto por la UCI	19
1.2.1 Definición.....	19
1.2.2 Objetivos.....	20
1.2.3 Importancia de la aplicación de los Lineamientos de Calidad de Software.....	20
1.2.3.1 Lineamientos de Calidad de Software	20
1.3 Expediente de Proyecto propuesto por la UCI.....	21
1.3.1 Definición.....	21
1.3.2 Situación actual de los Expedientes de Proyectos en la UCI	21
1.3.3 Historia del surgimiento del Expediente de Proyecto propuesto por la UCI.....	23
1.3.4 Necesidades propias de los proyectos de la UCI.....	25
1.3.5 Adaptación de normas y estándares al expediente definido para la UCI	26
1.3.6 Plantillas adicionadas al Expediente de Proyecto de la UCI.....	28
1.4 Conclusiones del Capítulo.....	28
CAPITULO 2: PROPUESTA DEL EXPEDIENTE DE PROYECTO.....	30
Introducción	30
2. Surgimiento del Expediente de Proyecto basado en Métrica 3	30
2.1 Estructura del Expediente de Proyecto basado en Métrica 3	34
2.1.1 Expediente de Proyecto para Proyectos de Desarrollo.....	34
2.1.2 Expediente de Proyecto para Proyectos de Servicios de Personalización	37
2.1.3 Expediente de Proyecto para Proyectos de Instalación.....	39
2.2 Métodos de Estimación de Métrica V3.....	40
2.2.1 Método Albrecht	40

2.2.1.1	Principales componentes del método Albrecht	40
2.2.1.2	Críticas del Método Albrecht.....	47
2.2.2	Método Mark II	47
2.2.2.1	Componentes del Método Mark II	48
2.2.2.2	Componente de Procesamiento en las Transacciones Lógicas.....	49
2.2.2.3	Pasos para calcular el Índice de Puntos Función	49
2.2.3	Comparación entre los Métodos Albrecht y Mark II	53
2.3	Roles y Responsabilidades de Métrica V3	57
2.3.1	Roles Principales.....	57
2.3.1.1	Perfil Directivo	57
2.3.1.2	Perfil Jefe de Proyecto.....	59
2.3.1.3	Perfil Consultor.....	61
2.3.1.4	Perfil Analista	62
2.3.1.5	Perfil Programador	65
2.4	Conclusiones del Capítulo.....	65
CAPITULO 3: APLICACIÓN Y ANÁLISIS DE RESULTADOS		67
Introducción		67
3.	Aplicación del Expediente de Proyecto de Instalación.....	67
3.1	Fase de Definición	67
3.2	Fase de Instalación	69
3.3	Fase de Pruebas	70
3.4	Grado de aceptación del Expediente de Proyecto de Instalación.....	71
3.5	Conclusiones	75
CONCLUSIONES GENERALES		76
RECOMENDACIONES		77
REFERENCIAS BIBLIOGRÁFICAS		78
BIBLIOGRAFÍA		82
ANEXOS.....		87
GLOSARIO DE TÉRMINOS		103

Figura 1. Procesos de la Metodología Métrica V3 [9].	15
Figura 2. Componentes del Método de Puntos Función de Albrecht [19].....	41
Figura 3. Ficheros Lógicos Internos (ILF) [19]	42
Figura 4. EO que referencia y deriva información [19].....	42
Figura 5. EQ con dos ILF y sin datos derivados [19].....	43
Figura 6. Componentes del Método de Puntos Función Mark II [20].....	48
Figura 7: Aceptación del Expediente de Proyecto	71
Figura 8: Orden Lógico del Plan de Implantación.....	72
Figura 9: Factibilidad del Expediente de Proyecto	73
Figura 10: Aceptación de las plantillas	74
Figura 11: Opinión del Expediente de Proyecto de Instalación	75

Tabla 1. Plantillas del Expediente de Proyecto.....	28
Tabla 2. Entradas Externas (EI).....	43
Tabla 3. EO y EQ	44
Tabla 4. Transacciones	44
Tabla 5. Tipo Registro	44
Tabla 6. ILF y EIF	45
Tabla 7. Cálculo de Puntos Función de Albrecht	45
Tabla 8. Características Generales del Sistema por Albrecht FPA	46
Tabla 9. Criterios para distinguir entidades primarias [20].....	50
Tabla 10. Encabezado de la tabla Transacciones Lógicas.....	51
Tabla 11. Características Generales del Sistema adicionales para MK II [20].....	52
Tabla 12. Participantes del perfil Directivo y actividades que realizan	59
Tabla 13. Participantes del perfil Jefe de Proyecto y Actividades que realizan	61
Tabla 14. Participantes del perfil Consultor y actividades que realizan.....	62
Tabla 15. Participantes del perfil Analista y actividades que realizan	65

INTRODUCCIÓN

En la actualidad el desarrollo de software se ha convertido en una de las más grandes perspectivas que tiene el mundo para su avance en la informática, por lo que es muy importante que este desarrollo se produzca de forma legible y eficiente.

Debido a esto nacen las metodologías de desarrollo de software, que no son más que el marco de trabajo de la ingeniería de software usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Con el uso de las metodologías el proceso se torna más cómodo, organizado, y con mayor facilidad de entendimiento para clientes y desarrolladores.

En el momento en que se decide comenzar el proceso de desarrollo de software, es donde aparecen las necesidades del usuario, las cuales son traducidas en requerimientos de software, conjuntamente se imponen las siguientes interrogantes: ¿Qué metodología de desarrollo de software usar?, ¿Cómo construir y organizar la documentación generada por esta metodología? Dichas interrogantes no son una tarea sencilla debido a la diversidad de propuestas metodológicas y diferencias en el grado de detalle, información disponible y alcance de cada una de las metodologías de desarrollo de software existentes.

Prueba de ello es que existen numerosas propuestas metodológicas dentro de ellas las metodologías estructuradas, estas se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán.

La Universidad de las Ciencias Informáticas (UCI) tiene definida una herramienta que agrupa y organiza todos los artefactos que se generan durante el desarrollo de software de los proyectos productivos, la misma lleva por nombre Expediente de Proyecto, este expediente está basado en Lineamientos de Calidad de Software, los cuales constituyen una guía a seguir por los proyectos de desarrollo de software atendiendo los artefactos que deben generar y procesos que deben realizar para aspirar a un producto final con calidad.

En el Área Temática Herramientas para la Teleformación de la UCI existen proyectos basados en programación estructurada, en los cuales el expediente propuesto por la UCI no es aplicable a la programación que utilizan, ya que este expediente está basado en metodologías orientadas a objetos.

Para darle solución a lo descrito anteriormente, se plantea el siguiente **problema científico**: ¿Cómo mantener el agrupamiento y organización de todos los artefactos que se generan durante el proceso de desarrollo de software de los proyectos basados en programación estructurada, en el Área Temática Herramientas para la Teleformación de la Universidad de las Ciencias Informáticas?

La presente investigación encierra como **objeto de estudio** las metodologías estructuradas. El **campo de acción** es la documentación generada por las metodologías estructuradas.

Para la formulación de la propuesta de expediente de proyecto basado en metodologías estructuradas se trazó el siguiente **objetivo general**: Proponer un Expediente de Proyecto basado en artefactos generados por una metodología estructurada para agrupar y organizar la documentación de los proyectos de desarrollo, servicio e instalación que utilizan programación estructurada del Área Temática Herramientas para la Teleformación.

Como **objetivos específicos** se definieron los siguientes:

- ❖ Identificar una metodología estructurada para guiar el proceso de desarrollo de software de los proyectos productivos que utilizan programación estructurada del Área Temática Herramientas para la Teleformación.
- ❖ Definir un Expediente de Proyecto que sea factible para la producción de software de los proyectos que utilizan programación estructurada de dicha Área Temática.
- ❖ Aplicar el expediente de proyecto dentro de algún proyecto basado en programación estructurada de dicha Área Temática y comprobar los resultados obtenidos.

En la investigación se plantea la siguiente **idea a defender**: La definición de un Expediente de Proyecto para los proyectos productivos basados en programación estructurada del Área Temática Herramientas para la Teleformación, permitirá agrupar y organizar todos los artefactos que se generan durante el proceso de desarrollo de software de estos proyectos.

Para dar cumplimiento a los objetivos se hace necesario desarrollar **tareas de investigación**, siendo éstas:

- ❖ Realización de un estudio de las metodologías estructuradas existentes.
- ❖ Estudio de los Lineamientos de Calidad de Software de la UCI.
- ❖ Análisis del Expediente de Proyecto propuesto por la UCI.
- ❖ Propuesta de la metodología estructurada a utilizar.
- ❖ Identificación de la documentación y productos que se generan para este tipo de metodología.
- ❖ Confección de un Expediente de Proyecto acorde con la metodología utilizada.
- ❖ Aplicación del Expediente de Proyecto definido a un proyecto del Área Temática Herramientas para la Teleformación que utilice programación estructurada.
- ❖ Realización de encuestas a los integrantes del equipo de proyecto donde se pondrá en práctica la propuesta de Expediente de Proyecto confeccionada.

Los métodos científicos que se utilizan para el desarrollo de la investigación son el Análisis y Síntesis para encontrar la relación entre los conceptos involucrados en las metodologías estructuradas, es decir analizar lo estudiado y sintetizar lo que llevamos a la práctica; el método Histórico-Lógico para investigar acerca de la historia de las metodologías estructuradas desde su inicio, aplicaciones que existen de las mismas en proyectos productivos tanto internacional como nacional, y en nuestra universidad. Otro método utilizado fue la encuesta, la misma fue de gran ayuda en la investigación ya que recopiló información acerca de la factibilidad de la propuesta de Expediente de Proyecto de Instalación, en cuanto a aceptación de las plantillas, y criterio de los integrantes del proyecto donde se realizó la instalación del software.

Capítulo 1: Fundamentación teórica. Se abordan de forma general los aspectos teóricos más importantes relacionados con las metodologías estructuradas, así como un estudio de los Lineamientos de Calidad de Software y Expediente de Proyecto definidos en la UCI para dar cumplimiento a los objetivos trazados en la presente investigación.

Capítulo 2: Propuesta del Expediente de Proyecto. Se realiza la propuesta de Expediente de Proyecto para tres tipos de proyecto, los proyectos de desarrollo, de servicio de personalización y los de instalación para agrupar y organizar todos los artefactos generados durante el proceso de desarrollo de software de los proyectos basados en programación estructurada del Área Temática Herramientas para la Teleformación.

Capítulo 3: Aplicación y análisis de resultados. Se aplica el Expediente de Proyecto propuesto a un proyecto productivo del Área Temática Herramientas para la Teleformación, y se analizan los resultados alcanzados con esta aplicación.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se describen algunos conceptos relacionados con las metodologías estructuradas necesarios para comprender el tema que se aborda en esta investigación. Se muestran los distintos tipos de metodologías estructuradas, sus características y estructura, así como los beneficios que reporta su utilización. También se presentan diferentes formas de clasificar las metodologías estructuradas, así como un estudio de los Lineamientos de Calidad de Software definidos por la UCI, y una investigación detallada acerca del surgimiento y características del Expediente de Proyecto propuesto por la misma.

1. Situación actual de las metodologías

A lo largo de los años han surgido gran cantidad de enfoques metodológicos respecto a la producción de software, pero muchas veces el contexto del proyecto es menos predecible o estable que lo deseado, lo que trae consigo que a nivel mundial se trabaje mayormente con la metodología RUP (Proceso Unificado de Rational), ya que ha conquistado a los procesos de software brindando mayor facilidad de uso a los usuarios. Una de las grandes facilidades de esta metodología es que brinda para su mejor documentación, los artefactos para cada una de las fases que la integran.

Existen otras metodologías igualmente conocidas pero menos utilizadas, estas son las metodologías estructuradas, las cuales definen los modelos del sistema que representan los procesos, los flujos y la estructura de datos de un modo descendente, pasando de una visión general del problema a un nivel de abstracción más sencillo.

Estas propuestas mayormente son aplicadas en grandes empresas y proyectos que ponen en práctica metodologías orientadas a procesos y a datos, por ejemplo en la facultad de Ingeniería Mecánica y Eléctrica de la Universidad de la Colima, en México, el Sistema de Nómina está basado en la metodología

de Yourdon.

Al inicio de la informática en Cuba no existía cultura de industria en las empresas de software hasta hace menos de una década, lo que trajo consigo que no se fabricara software comercial sino artesanal por completo. Antes del año 2000 quienes utilizaban metodologías en Cuba eran los centros adscritos a universidades o las propias universidades como la CUJAE que utilizaban: (ADESA), Análisis y Diseño Orientado a Objetos de Sistemas Informáticos, (ADOOSI) para sistemas de gestión estructurados u Orientado a Objeto, y MultiMet para sistemas hipermedia en general.

Fue sólo a partir del surgimiento de RUP que se comenzó con la diversificación en el sentido de las metodologías. Contribuyó a ello el interés político del país de pasar de artesanos a industriales en la producción de software lo que hizo que nacionalmente se comenzaran a interesar por este tema todos aquellos que producían software. Es por ello que Cuba se dio la tarea de investigar a fondo las diversas metodologías de desarrollo de software, y cómo aplicarlas a proyectos de sus mismas características, para así garantizar una fácil gestión del proceso de desarrollo de un software, que le facilitara al cliente un mejor entendimiento del mismo.

Por tales causas se puede asegurar que las empresas adscritas al MIC (Ministerio de la Informática y las Comunicaciones) se inclinan por el uso de RUP, pero con ciertas modificaciones para contextualizarlo a sus intereses. Además, en las universidades, especialmente en la UCI, la formación docente en la materia que incluye a las metodologías de desarrollo, están sustentadas sobre la base de la metodología RUP ya que es la más completa, factible, robusta y la más conocida de todas, pero también se utilizan metodologías estructuradas en algunos proyectos que implementan sus aplicaciones con programación estructurada, estas metodologías no son tan conocidas como RUP pero también tienen sus ventajas, como por ejemplo, Intentan aplicar formas ingenieriles para solucionar problemas técnicos al obtener un sistema de información. [1]

Esto da la medida de la poca usabilidad y conocimientos de los procesos estructurados. Es por ello que se lleva a cabo esta investigación, con el objetivo de profundizar más a fondo los conocimientos sobre las

metodologías estructuradas en la UCI, y enfatizar aún más en la manera más fácil de documentarla, es decir, definir una guía organizativa que plasme todos los procesos, y funcionalidades tanto del proyecto como de la metodología estructurada utilizada en el mismo.

1.1 Metodologías Estructuradas

1.1.1 Historia

Hasta fines de los años 70, el analista escribía lo que entendía de los requerimientos del usuario en un enorme documento que consistía primariamente en una narrativa. Esto se veía afectado por diversos problemas importantes, como por ejemplo: eran monolíticos, es decir había que leer completamente la especificación, de principio a fin, para poder entenderla, además eran redundantes, ambiguas e imposibles de mantener. [2]

Mientras se debatían todos estos problemas, ya se estaba adaptando un conjunto complementario de ideas en el área de programación y diseño, normalmente conocidas como diseño y programación estructurados, que prometían grandes mejoras en la organización, codificación, prueba y mantenimiento de los programas de computadora. La programación estructurada mantiene el criterio que la elaboración de programas de computador es una labor que demanda esfuerzo, creatividad, habilidad y cuidado. Como resultado de la programación estructurada surgió un movimiento gradual a hacer especificaciones funcionales que fueran: gráficas, particionadas y mínimamente redundantes. [3]

Este enfoque que se conoce hoy como estructurado, se utiliza ahora en la mayoría de las organizaciones de desarrollo de sistemas orientados a los negocios, procesos, y un gran número orientadas hacia la ingeniería. Dicho enfoque también abarca el mundo de las metodologías de desarrollo de software, las cuales poseen tres grandes generaciones, como son:

Desarrollo convencional: Se caracteriza porque los resultados finales son impredecibles, además no hay forma de controlar lo que está sucediendo en el proyecto.

Desarrollo orientado a objetos: La esencia del desarrollo orientado a objetos es la identificación y organización de conceptos del dominio de la aplicación. Además, son iterativas e incrementales y fáciles de dividir el sistema en varios subsistemas independientes.

Desarrollo estructurado: Dentro del desarrollo estructurado se encuentra la programación estructurada, o también como se le solía llamar: “*Técnicas para el mejoramiento de la productividad en programación*”, que no es más que una forma de escribir programación de computadora de forma clara. Para ello se utilizan únicamente tres estructuras: **secuencial**, **selectiva** e **iterativa**, siendo innecesario y no permitiéndose el uso de la instrucción o instrucciones de transferencia incondicional, surgiendo también el diseño y análisis estructurado, además de las especificaciones funcionales antes mencionadas. [2]

Mediante la programación estructurada todas las bifurcaciones de control de un programa se encuentran estandarizadas, de forma tal que es posible leer la codificación del mismo desde su inicio hasta su terminación en forma continua, sin tener que saltar de un lugar a otro del programa siguiendo el rastro de la lógica establecida por el programador.

El resultado de aplicar programación estructurada es obtener una alta precisión nunca antes lograda, además que las pruebas que se le realizan a los programas desarrollados utilizando este método, se acoplan más rápidamente y el resultado final con programas que pueden ser leídos, mantenidos y modificados por otros programadores con mucha mayor facilidad. “*La programación Estructurada está basada en el **Teorema de la Estructura**, el cual establece que cualquier programa propio (un programa con una entrada y una salida exclusivamente) es equivalente a un programa que contiene solamente las estructuras lógicas mencionadas anteriormente.*” [4]

Como una posibilidad de “*nuevo orden*” [1], surge el desarrollo estructurado, sobre la base de la programación estructurada, los métodos de análisis y el diseño estructurado. Esta nueva etapa es la piedra fundamental para la construcción de programas con métodos ingenieriles. La programación estructurada,

pasa al ámbito empresarial en los setenta. Tiene como punto de partida el establecimiento y uso de normas para la aplicación de estructuras de datos y control.

La base de la programación y el diseño estructurado, es un análisis del problema usando el diseño “*top-down*” o *descendente* [1], con énfasis en las especificaciones funcionales. Se compone de diagramas, con textos de referencia de los mismos, y con independencia para que se puedan leer en forma parcial, con una redundancia mínima, de modo que los cambios no lo afecten en forma notable.

1.1.1.1 Clasificación

Las metodologías estructuradas se clasifican en [1]:

- ❖ **Orientadas a procesos**
- ❖ **Orientadas a datos**
 - ✓ Estructura de datos jerárquicos
 - ✓ Estructura de datos no jerárquicos
- ❖ **Mixtas**
- ❖ **Metodologías orientadas a procesos:** Se apoyan en técnicas gráficas para obtener especificación estructurada.
- ❖ **Metodologías orientadas a datos jerárquicos:** La estructura de control del programa debe ser jerárquica y se debe derivar de la estructura de datos del programa.
- ❖ **Metodologías orientadas a datos no jerárquicos:** Planificación, análisis, diseño y construcción.
- ❖ **Metodologías mixtas:** Estudian los sistemas desde varios puntos de vista.

1.1.1.2 Características

Entre las características que presentan las metodologías estructuradas se encuentran las siguientes [1]:

- 1- Definen los modelos del sistema que representan los procesos, los flujos y la estructura de datos de un modo descendente, pasando de una visión general del problema a un nivel de abstracción más sencillo.
- 2- Proponen la creación de modelos del sistema.

3- Tiene un enfoque Top- Down.

4- Se apoyan en técnicas gráficas para obtener una especificación estructurada.

1.1.1.3 Ejemplos de Metodologías Estructuradas

A nivel mundial existen varias metodologías para el desarrollo de software que encajan bajo el estandarte de estructuradas. Dentro de las más conocidas citar [5]:

- ❖ Metodología de Yourdon/Constanstine
- ❖ Metodología de DeMarco (Merise)
- ❖ Metodología de Gane & Sarson
- ❖ Metodologías Orientadas a Datos (Jackson, Warnier-Orr)
- ❖ SSADM
- ❖ EUROMÉTODO
- ❖ Métrica V3

A continuación una breve explicación de cada una de estas metodologías.

1.1.1.3.1 Metodología de Yourdon/Constanstine

Las metodologías para Yourdon-Constanstine, tienen una fuerte orientación y exposición a los procesos, hacen hincapié en la dinámica del flujo de datos. Un fenómeno interesante fue la aparición a finales de los años 70 y principios de los 80, de una reimpresión de las técnicas de proyecto estructurado, como las obras de Yourdon-Constantine (1979), Higgins (1979), Page-Jones (1980), Stevens (1981) y otros [5].

1.1.1.3.2 Metodología de DeMarco (Merise)

Las bases de MERISE comenzaron en 1972 por un equipo universitario de ingenieros de Aix-en-Provence. La primera versión salió a finales de 1976. El proyecto partió del Centre Technique Informatique del Ministerio de Industria Francés en septiembre de 1977, para cubrir las necesidades tanto de la administración como de las empresas. El proyecto finalizó en mayo de 1978 dando lugar a MERISE como metodología de Análisis y Diseño de Sistemas de Información. Esta metodología aporta un ciclo de vida más largo a los existentes hasta entonces que se materializa en un conjunto definido de etapas. Introducen dos ciclos complementarios: ciclo de abstracción y ciclo de decisión. El ciclo de abstracción se basa en la percepción de tres niveles de abstracción: conceptual, organizativo y físico. Además se definen dos niveles para cada nivel: un modelo de datos y otro de tratamientos [5].

1.1.1.3.3 Metodología de Gane & Sarson

La herramienta básica de estas metodologías es el diagrama de flujo de datos (DFD). Aunque esta forma de representación es extremadamente simple y comprensible que la descripción de casos en lenguaje natural, todavía tiene muchas limitaciones. Por ejemplo, no hay manera de representar la información de control. A pesar de estar relacionado el nombre de la herramienta con "datos", de hecho, el énfasis está en los archivos. Tampoco hay reglas claras para que detalle la medida del DFD. Dependiendo de la experiencia del desarrollador y su conocimiento del problema, estas especificaciones pueden ser insuficientes y en otros casos, excesivo. Gane-Sarson proponen el uso del modelo de datos adicionales, como una manera de dar una mayor claridad conceptual para el proceso de desarrollo de software. Esto permite incluso tratar los problemas de tamaño significativamente mayor, difícil de tratar solamente el nivel de DFD y Diccionario de Datos. [6]

1.1.1.3.4 Metodologías orientadas a datos (Jackson, Warnier-Orr)

Lo más fuerte de la metodología de Jackson es su orientación para la estructura de datos. En la mayoría de los proyectos, los procedimientos son bastante sencillos, mientras que las estructuras de datos pueden ser mucho más complejas. Por otra parte, Jackson no da importancia al proyecto de la lógica de control,

siempre y cuando sea impuesto por estructuras de datos. Rectores de su metodología para los programas simple (o complejo programa descompuesto hasta que pueda ser considerada como un simple programa). También se da un cierto nivel de especificación con anterioridad, lo cual no siempre es cierto. Por otra parte, la metodología está más centrada en una vista jerárquica de estructuras de datos, cada vez menos común en los sistemas en línea, basado en los bancos de datos.

La metodología Warnier-Orr es bastante innovadora en relación con las demás, teniendo su base en la lógica de juegos de matemáticas. Ella sale de la especificación del sistema, para deducir los procesos y la introducción de datos, y su diagrama, que separa adecuadamente aspectos lógicos y físicos del sistema. Sin embargo, este diagrama se extiende por lo general a lo particular, de una forma prácticamente lineal (de derecha a izquierda), es realmente difícil de representar a los sistemas de cierta consideración. Pero la principal limitación de esta metodología es su insuficiencia para representar las estructuras de datos que no son jerárquicas, como los sistemas actuales de varias bases de datos. Esto probablemente favorece la construcción de sistemas orientados a la emisión de informes y consultas, pero no parece adecuado a los sistemas más complejos [5].

1.1.1.3.5 SSADM

Esta metodología nace como necesidad del gobierno británico, la cual se desarrolló entre el Central Computing and Telecommunications Agency (CCTA) y Learmonth and Burchett Management Systems (LBMS), dando como resultado la metodología SSADM (Structures Systems Analysis and Design Method). La misma hace énfasis en los usuarios y sus requisitos, además define el proceso de producción, posee máxima flexibilidad en herramientas y técnicas de implementación, y tiene tres puntos de vista: datos, eventos y procesos. SSADM proporciona un conjunto de procedimientos para llevar a cabo el análisis y diseño, pero no cubre aspectos como la planificación estratégica ni entra en la construcción del código [6].

1.1.1.3.6 EUROMÉTODO

Eurométodo es una metodología para la adquisición de sistemas de información y servicios relacionados, desarrollada en el marco de un proyecto del mismo nombre, bajo supervisión de la Comisión Europea. La versión en español de EUROMÉTODO v1, la más reciente, ha sido publicada en 1998. Como marco metodológico proporciona un conjunto de conceptos y una terminología para mejorar la relación cliente-proveedor. Por otro lado como método sirve para definir, planificar y ejecutar la adquisición de un sistema de información y los servicios relacionados [7].

1.1.1.3.7 Métrica 3

Métrica v3 es una metodología de desarrollo de distribución y uso libre, elaborada por el Consejo Superior de Informática del Ministerio de Administraciones Públicas Español. Métrica tiene ya varios años de vida y su actual versión, la 3, se crea con la finalidad de incorporar las nuevas técnicas derivadas de la programación y el análisis orientado a objetos, al proceso de desarrollo de software, que a través de esta metodología las administraciones públicas españolas pretenden llevar a cabo.

En una única estructura la metodología MÉTRICA Versión 3 cubre distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organización: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad. La automatización de las actividades propuestas en la estructura de MÉTRICA Versión 3 es posible ya que sus técnicas están soportadas por una amplia variedad de herramientas de ayuda al desarrollo [8].

Objetivos

Métrica 3 tiene sus bases en los siguientes objetivos [8]:

- ❖ Mejorar de la productividad del desarrollador al planificar claramente sus pasos.
- ❖ Incrementar la robustez del producto final.
- ❖ Minimización de los riesgos.

- ❖ Facilitar la comunicación entre todos los miembros del equipo de desarrollo.
- ❖ Sencilla, práctica y de fácil uso.
- ❖ Formal: establece los pasos necesarios.
- ❖ Abierta: apta para administración y empresas.

Características

Métrica 3 hace posible gestionar proyectos orientados a objetos y estructurados, y ofrece un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software [9].

- ❖ Es una metodología de planificación y sistemas de información.
- ❖ Hace posible gestionar proyectos orientados a objetos y proyectos estructurados.
- ❖ Es práctica, actual y recoge lo mejor de otras metodologías como SSADM y Merise.
- ❖ Es flexible y aplicable a proyectos con diferentes ciclos de vida.
- ❖ Incluye arquitectura cliente servidor desde el principio.
- ❖ En diseño incorpora la arquitectura de soporte como clave para la reutilización.
- ❖ Está basada principalmente en el ciclo de vida principal o de cascada.
- ❖ Distingue:
 - ✓ Procesos principales: (planificación, desarrollo, y mantenimiento).
 - ✓ Procesos de interfaz: (calidad, seguridad, gestión y configuración).

Procesos

Todo proyecto que utilice Métrica V3 consta de un conjunto de procesos o fases que se desglosan en múltiples puntos, cuya cronología hay que seguir con claridad para ir avanzando en el desarrollo del proyecto. El siguiente gráfico muestra el esquema general de estos procesos:

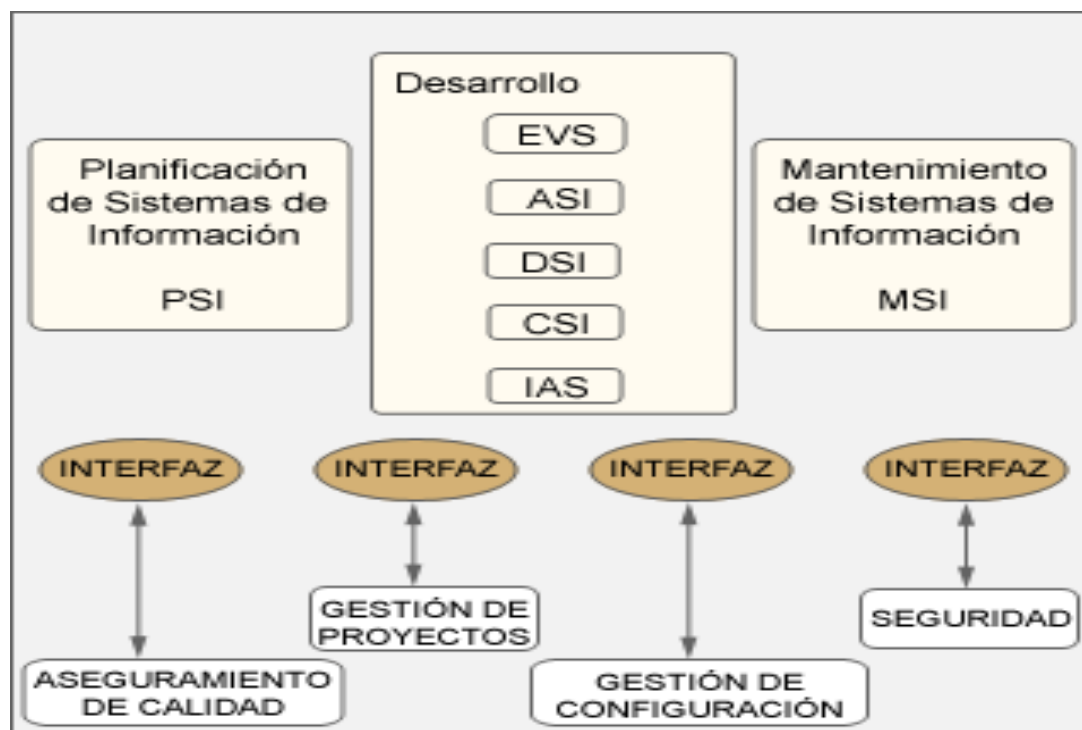


Figura 1. Procesos de la Metodología Métrica V3 [9].

El punto de partida de Métrica V3 es la versión anterior, de la cual se ha observado la adaptabilidad, flexibilidad y sencillez, así como la estructura de actividades y tareas, las fases y módulos de MÉTRICA versión 2.1 han dado paso a la división en procesos, más adecuada a la entrada-transformación-salida que se produce en cada una de las divisiones del ciclo de vida de un proyecto [9].

- ❖ Proceso 1: Planificación del Sistema de Información
- ❖ Proceso 2: Desarrollo de Sistemas de Información
 - ✓ Estudio de Viabilidad del Sistema
 - ✓ Análisis del Sistema de Información
 - ✓ Diseño del Sistema de Información
 - ✓ Construcción del Sistema de Información
 - ✓ Implantación y Aceptación del Sistema

❖ Proceso 3: Mantenimiento de Sistemas de Información

A continuación una breve descripción de estos procesos.

❖ **Proceso 1: Plan de Sistemas de Información (PSI)**

En este proceso de la metodología de Métrica V3 proporciona un marco estratégico de referencia para los Sistemas de Información de un determinado ámbito de la organización y además define [9]:

- ✓ La información necesaria para satisfacer los objetivos estratégicos de la organización.
- ✓ La arquitectura de la información, en cuanto a procesos y datos, y así satisfacer los objetivos estratégicos de la organización.
- ✓ Los nuevos sistemas a desarrollar para implantar dicha arquitectura.

❖ **Proceso 2: Desarrollo de Sistemas de Información**

Este proceso abarca todas las actividades y tareas necesarias para el desarrollo del software, desde el análisis de requisitos hasta la instalación del software. Dada su complejidad se subdivide a su vez en varios subprocesos [9].

✓ **Estudio de Viabilidad del Sistema (EVS)**

El objetivo del Estudio de Viabilidad del Sistema es el análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo, que tenga en cuenta restricciones económicas, técnicas, legales y operativas. La solución obtenida como resultado del estudio puede ser la definición de uno o varios proyectos que afecten a uno o varios sistemas de información ya existentes o nuevos. Para ello, se identifican los requisitos que se ha de satisfacer y se estudia, si procede, la situación actual [9].

✓ **Análisis del Sistema de Información (ASI)**

El objetivo de esta fase es describir el alcance y los requisitos del sistema generando diferentes alternativas para solucionar el problema y recomendar una de ellas si es oportuno. Una vez seleccionada una alternativa se tiene que generar las especificaciones formales que describan al sistema y estas tienen que ser aprobadas por el usuario, y además pretende conseguir una especificación detallada del Sistema de Información a través de un catálogo de requisitos y una serie de modelos [9].

✓ **Diseño del Sistema de Información**

Este proceso tiene como objetivo obtener las especificaciones físicas del sistema para la construcción del mismo. Se describe cómo será el sistema desde el punto de vista físico, para ello se debe diseñar la arquitectura física del sistema, el esquema externo de datos del sistema teniendo en cuenta el entorno tecnológico en que va a funcionar [9].

✓ **Construcción del Sistema de Información**

El objetivo de este proceso es construir y probar los componentes del sistema obtenidos en las especificaciones físicas de la fase anterior de Diseño de Sistemas. Se realizan la construcción, pruebas unitarias y de integración del equipo lógico y se desarrollan los procedimientos de operación de componentes del sistema, además de crearse los manuales de usuario [9].

✓ **Implantación y Aceptación del Sistema**

El objetivo de esta fase es conseguir la aceptación final del nuevo sistema por parte de los usuarios y poner en funcionamiento el nuevo sistema. Se realizarán las pruebas de sistema y las de aceptación, es decir, se comprueba el sistema en su totalidad y así posteriormente, se pone en explotación [9].

❖ **Proceso 3: Mantenimiento de Sistemas de Información**

Según ISO el mantenimiento comprende la modificación o retirada de todos los componentes del Sistema de Información pero al estar Métrica orientado al proceso de desarrollo solo refleja los aspectos de mantenimiento correctivo y evolutivo relacionados con el desarrollo (solución de errores y mejora añadiendo funciones) [9].

Interfaces

Métrica 3 define un conjunto de actividades de tipo organizativa o de soporte al proceso de desarrollo y productos, entre ellas [9].

- ❖ **Gestión de Proyecto (GP):** Planificación, seguimiento y control de actividades y recursos humanos y materiales.
- ❖ **Seguridad (SG):** Análisis de riesgos lógicos.
- ❖ **Gestión de la Configuración (GC):** Definición y control de los cambios en la configuración del sistema, modificaciones y versiones.
- ❖ **Aseguramiento de la Calidad (AC):** Marco de referencia para la definición y puesta en marcha de planes de aseguramiento de la calidad.

Aportaciones

Para un mejor entendimiento de la metodología Métrica 3, la misma propone una serie de aportaciones [9].

- ❖ **Compatibilidad con los últimos estándares en calidad e ingeniería del software.**
- ❖ **Interfaz con procesos y gestión de control.**
- ❖ **Cubre desarrollos estructurados y orientados a objetos.**
- ❖ **Ampliación para tecnologías cliente\servidor e interfaces de usuarios gráficas (GUI).**
- ❖ **Técnicas soportadas por varias herramientas (CASE).**

Argumentos del uso de la Metodología Métrica 3

Después de haber caracterizado cada una de las metodologías, es fácil deducir que el uso de procedimientos estructurados depende de diferentes factores, y son extensamente aplicables y deben ser usados por más personas de las que actualmente los tienen en cuenta.

Todas las metodologías antes mencionadas son de gran importancia para guiar el desarrollo de software estructurado, pero hay una de ellas que se destaca mucho más debido a su amplia visión estructurada, Métrica V3. Esta metodología fue seleccionada con el propósito de organizar y guiar el proceso de desarrollo de software de los proyectos que utilizan programación estructurada en el Área Temática Herramientas para la Teleformación, es por ello que la presente investigación, aborda más a fondo las fases, objetivos, interfaces, características, y procesos de Métrica V3, lo cual conllevará a conocer sus facilidades de uso, y a la vez proponer un Expediente de Proyecto basado en la misma que organice y agrupe todos los artefactos que se generan durante el proceso de desarrollo de software de los proyectos anteriormente mencionados.

Otras de las razones por las cuales se decidió utilizar Métrica V3 fue por las semejanzas que presenta con RUP, y por la facilidad de entendimiento que poseen sus fases y procesos.

1.2 Lineamientos de Calidad de Software propuesto por la UCI

1.2.1 Definición

Los Lineamientos de Calidad de Software constituyen una guía a seguir por los proyectos de desarrollo de software atendiendo los artefactos que deben generar y procesos que deben realizar para aspirar a un producto final con calidad [10].

1.2.2 Objetivos

Los Lineamientos de Calidad de Software tienen como objetivo [10]:

- ❖ Garantizar la uniformidad de la documentación generada por los proyectos productivos.
- ❖ Definir reglas de estricto cumplimiento, que garantizan la facilidad de entendimiento del software.

1.2.3 Importancia de la aplicación de los Lineamientos de Calidad de Software

Los Lineamientos de Calidad de Software son un conjunto de procedimientos a seguir para la realización de documentos formales de la universidad, como por ejemplo, el Expediente de Proyecto; dichos lineamientos priorizan la uniformidad de la documentación generada por las metodologías de desarrollo de software utilizadas en los proyectos productivos.

Uno de los pasos más importantes que posee el proceso de desarrollo de software en la UCI, es la documentación del mismo, y estos lineamientos propuestos por la universidad que constituyen un pequeño conjunto de prácticas básicas para orientar la producción de software, garantizan un paso a seguir en esa documentación, mas no garantizan con su aplicación productos de alta calidad, pues se trata de un pequeño grupo de elementos para empezar.

Estos lineamientos son ajustados a cada caso particular, y además son aplicables a cualquier proyecto de desarrollo de software de la UCI, permitiendo la orientación y organización del trabajo dentro y fuera del proyecto.

Los mismos se complementan con: una herramienta de diagnóstico (Propuesta) en forma de encuesta que sirve como guía para revisar su cumplimiento y con una estrategia de Desarrollo y Liberación de un Producto Software como resumen general del proceso [10].

1.2.3.1 Lineamientos de Calidad de Software

Se llevó a cabo un estudio detallado de los Lineamientos de Calidad de Software, garantizando la uniformidad de las planillas que conforman el Expediente de Proyecto, de los proyectos basados en programación estructurada del Área Temática Teleformación perteneciente al Polo Software Educativo de la UCI.

Estos lineamientos que exige calidad nos dan la medida, de lo estricta que puede llegar a ser la documentación de un software, así como el gran número de requisitos que conllevan al desarrollo del mismo. Es por ello que siempre se recomienda documentar, organizar y controlar el proceso de desarrollo del software, para así lograr establecer una mejor comunicación con el cliente o usuario [10].

1.3 Expediente de Proyecto propuesto por la UCI

1.3.1 Definición

El Expediente de Proyecto es la herramienta que agrupa y organiza todos los artefactos que se generan durante el proceso de desarrollo de software de los proyectos productivos.

1.3.2 Situación actual de los Expedientes de Proyectos en la UCI

La Universidad de las Ciencias Informáticas se ha convertido, en muy poco tiempo, en centro de referencia para la Industria Cubana del Software. En apenas 5 años sorprendería ver el número de proyectos que han nacido y crecido en la misma. El modelo productivo que caracteriza la producción en la UCI es otro elemento distintivo.

La vinculación formación-producción-investigación debe tener en cuenta todo modelo, procedimiento y plantilla que se defina. Todo esto ha impuesto un reto en cuanto a la organización de la producción, apoyados en la definición de calidad de Pressman [11], “concordancia de los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y de las características implícitas que se espera de todo software desarrollado profesionalmente” y en aras

de lograr la calidad tanto en los procesos como en los productos se ha trabajado en la estandarización de los procesos de desarrollo de software y la documentación que en estos se generan.

En cualquier caso sería evidente la necesidad de querer documentar suficientemente el producto y su proceso de producción. Por tanto, desde que un proyecto se inicia debe abrirse un expediente de proyecto que recoja toda la documentación necesaria para garantizar la calidad del proceso de producción del software como del producto en sí.

En la actualidad no existe una homogeneidad en la documentación que se construye por parte de los proyectos, lo que dificulta la retroalimentación que pudiera obtenerse de la misma, así como su reutilización por parte de otros proyectos.

Como parte de los mecanismos que permitan mejorar la calidad de los procesos de desarrollo de software, estandarizar la documentación de todos los proyectos que se realizan, facilitar la formación y adiestramiento de los equipos de proyecto en el uso de modelos y estándares propios y para elevar la calidad del producto entregado al cliente, se ha diseñado un Expediente de Proyecto que se deberá implantarse en cada uno de los proyectos vigentes en la universidad. Este esquema de expediente y grupo de plantillas tipo definidas tienen como objetivo influir en la estandarización de la documentación y la creación de una cultura de calidad en la organización.

La documentación asociada a los proyectos de software y sistemas debe cumplir con algunos requisitos como son [10]:

- ❖ Servir como medio de comunicación entre los miembros del equipo.
- ❖ Servir de repositorio de información que pueda ser utilizado por los ingenieros de sistemas.
- ❖ Proveer información para el control de los planes, cronogramas e hitos en el proceso de desarrollo de software.
- ❖ Definir quién hace y cómo hace las actividades específicas del desarrollo.

Satisfacer estos requerimientos requiere de un grupo de documentos que no son solamente las documentaciones técnicas, asociadas al producto. No es posible organizar la documentación de los proyectos sin definir un esquema genérico para la organización que norme como usar y evaluar la documentación según el tipo de proyectos.

Para lograr esta organización en los proyectos de desarrollo de software en la UCI se decidió definir una estructura para el Expediente de Proyecto y un grupo de plantillas para este expediente. Para esto fueron analizados varios modelos y estándares que norman la documentación de software como el Modelo de Madurez y Capacidades (CMMI) [12] en su versión 1.2, las normas ISO y los estándares de la IEEE [13] [14]. Otro aspecto a tener en cuenta es la compatibilidad que debía tener dicho expediente con el modelo CMMI modelo seleccionado para los procesos de desarrollo de software en la Universidad. [10]

1.3.3 Historia del surgimiento del Expediente de Proyecto propuesto por la UCI

En los años 80s tratando de dar solución a las crisis del software, el Departamento de Defensa de los Estados Unidos creó el Instituto e Ingeniería de Software (SEI) en la Universidad Carnige Mellow.

Desde sus inicios el SEI se enfrascó en lograr una estandarización en los modelos a utilizar para la construcción de software. Muestra de ello es la creación del Proceso de Software Personal (PSP) [15], el Proceso de Software de Equipo (TSP) y en 1991 del Modelo de Madurez y Capacidades (CMM) [16]. Este último pretendía implantar un modelo de procesos por el cual medir la madurez y la capacidad de una empresa productora de software. El modelo fue evolucionando hasta que, en 2001, el SEI lanza el Modelo de Madurez y Capacidades Integrado (CMMI). [10]

CMMI divide sus Áreas de Procesos (AP) en 3 grupos:

- ❖ Ingeniería
- ❖ Soporte
- ❖ Gestión de proyecto

Estas mismas clasificaciones fueron las utilizadas para agrupar las plantillas en el Expediente de Proyecto (EP) agregando una cuarta categoría para la documentación legal que, dada su naturaleza y lo delicado de la misma no se consideró conveniente mezclar con documentación técnica.

La definición e institucionalización del Expediente de Proyecto se realizó siguiendo el modelo escalonado definido CMMI por cada uno de sus niveles de madurez iniciando por el nivel 2 incluyendo algunas exigencias del nivel 3.

De esta manera es como termina conformado el Expediente de Proyecto, el mismo consta de tres grupos contenedores de las planillas, que son las que recogen la información de cada uno de los procesos de desarrollo de software. A continuación, una explicación, acerca de la información que recogen estos grupos:

Ingeniería

En este grupo se encuentran todos aquellos documentos que se encuentran relacionados con los procesos de la Ingeniería de Software. Su construcción no está basada en ninguna metodología en específico, aunque sí se tienen en cuenta los procesos básicos para la construcción de software.

Al ser lo más genérico posible hay metodologías que pueden no estar satisfechas totalmente con la documentación, como pueden ser el caso de las metodologías para aplicaciones multimedia y Extreme Programming, lo cual no quiere decir que no puedan aplicarse muchas de las plantillas presentadas. Este aspecto quedará solucionado para la próxima versión del expediente.

Gestión de proyecto

Este grupo recoge los documentos necesarios para la administración efectiva de un proyecto software. Esto se traduce en los planes para la ejecución de cada uno de los procesos presentes en el desarrollo de

software así como aquellos aspectos generales del proyecto (cronogramas, recursos, presupuestos, visión del proyecto...).

Soporte

En este grupo se encuentran los documentos que garantizan el soporte al software en construcción y se encuentra formado por dos subgrupos: Gestión de Configuración de Software y Aseguramiento de la Calidad [10].

1.3.4 Necesidades propias de los proyectos de la UCI

La UCI se dio la tarea de identificar sus propias necesidades en cuanto a la adaptabilidad del expediente de proyecto que iba a poner en práctica, uno de los pasos que se tuvieron en cuenta fueron todos los estándares y normas precisas para adaptar el expediente de proyecto a una metodología de software específica. Para esto se tomó como referencia el Rational Unified Process [17] dada su adaptabilidad a casi cualquier proceso de desarrollo que se quiera utilizar y adaptable a una gran variedad de tipos de proyectos. El próximo paso lo constituye adaptar el modelo CMMI a dicha metodología.

Las principales diferencias se encontraron en las AP de ingeniería, específicamente en las actividades de análisis y diseño e implementación.

Otra área que se vio necesario reforzar fue la referente a la arquitectura, en dos aspectos: la arquitectura del software y la arquitectura de información.

La UCI, como prospecto de la nueva universidad cubana tiene un modelo compuesto por tres procesos fundamentales: Producción – Formación – Investigación, y este modelo ha sido uno de los retos de este expediente, lograr sincronizar estos tres procesos. En esto tiene gran impacto el área de Entrenamiento Organizacional (OT) de CMMI. Esta es un área de proceso de Nivel 3 de CMMI, pero a pesar de que este expediente está basado en el Nivel 2 de dicho modelo, se tuvieron en cuenta algunos elementos dentro del

Plan de Capacitación. Teniendo en cuenta la importancia de la integración de estos tres procesos para la UCI, esta área de proceso se verá reforzada en la próxima versión.

El resto de las áreas de procesos eran perfectamente compatibles por lo que se tomó la decisión de dejarlas tal y como estaban propuestas dentro del expediente [10].

1.3.5 Adaptación de normas y estándares al expediente definido para la UCI

Para cada área tenida en cuenta en el expediente se definieron plantillas que permitirán estandarizar la documentación de los proyectos de la UCI. La selección de las plantillas se realizó a partir de la revisión de las propuestas por la IEEE [13] [14], la NASA en sus Sistema de Gestión de la Calidad, las normas ISO relacionadas con la producción de software, DOD y RUP.

Partiendo de este análisis y de la experiencia personal de líderes de proyectos y desarrolladores el expediente quedó conformado por las plantillas que se muestra en la tabla 1 [10].

Ingeniería	Gestión de proyecto	Soporte	Legales
Plantilla DCS - Diagrama de Proceso - Nombre del Proceso v1.0	Plantilla DCS - Plan Desarrollo de Software v1.0	Plantilla DCS - Glosario de Términos	Plantilla ALBET - Acta de Aceptación
Plantilla DCS - Especificación de Requisitos v1.0	Plantilla DCS - Presupuesto v1.0	Plantilla DCS - Listas de Chequeo	Plantilla ALBET - Acta de Entrega
Plantilla DCS - IDEF0 Diagram Shapes	Plantilla DCS - Lista de Riesgos v1.0	Plantilla DCS - No Conformidades (ampliada) v1.0	Plantilla ALBET - Acta de inicio de proyecto
Plantilla DCS - Modelo de Casos de uso del sistema v1.0	Plantilla DCS - Plan Mitigación de Riesgos v1.0	Plantilla DCS - No Conformidades (reducida) v1.0	Plantilla ALBET - Acta de Terminación de Proyecto

Plantilla DCS - Plan de Gestión de Requisitos v1.0	Plantilla DCS - Ambiente de Desarrollo v1.0	Plantilla DCS - Plan Aseguramiento de la Calidad v1.0	Plantilla ALBET - Carta
Plantilla DCS - Especificación de Requisitos v1.0	Plantilla DCS - Plan de Capacitación	Plantilla DCS - Plan de Mediciones v1.0	Plantilla ALBET - Indefiniciones
Plantilla DCS - Modelo del Negocio v1.0	Plantilla DCS - Roles y Responsabilidades v1.0	Plantilla DCS - Respuestas a No Conformidades v1.0	Plantilla ALBET - Informe Técnico
Plantilla DCS - Modelo del Dominio v1.0	Plantilla DCS - Documento Visión v1.0	Plantilla DCS - Solicitud de Cambio (reducida) v1.0	Plantilla ALBET - Minuta de Reuniones
Plantilla DCS - Arquitectura de Información v1.0	Plantilla DCS - Diagnóstico v1.0	Plantilla DCS - Pedido de Cambio v1.0	Plantilla ALBET - Proyectos Técnicos
Plantilla DCS - Documento de Arquitectura de Software v1.0	Plantilla DCS - Minutas de Reuniones v1.0	Plantilla DCS - Plan Gestión de Configuración v1.0	
Plantilla DCS - Informe del Levantamiento de Información para la Arquitectura de Información v1.0			
Plantilla DCS - Modelo de Diseño v1.0			
Plantilla DCS - Diseño Casos de Prueba v1.0			

Plantilla DCS - Plan de Pruebas v1.0			
Plantilla DCS - Modelo de Despliegue v1.0			

Tabla 1. Plantillas del Expediente de Proyecto.

Un paso importante en la confección del Expediente de Proyecto lo fue la adaptación de las plantillas seleccionadas al entorno de la UCI. Para esto se utilizó la terminología manejada por los proyectos, se describieron los epígrafes y algunos nombres fueron adaptados a los artefactos manejados por los proyectos.

1.3.6 Plantillas adicionales al Expediente de Proyecto de la UCI

Como se mencionó en epígrafes anteriores, en la UCI se construyen artefactos que no se encuentran formalmente en las normas y estándares internacionales consultados, por lo que fue necesario realizar una revisión de las variantes de los mismos, utilizadas por diferentes proyectos para así llegar a una plantilla fácil de asimilar por todos los proyectos [10].

Ejemplos de estos documentos son:

- ❖ Plantilla DCS - Evaluación de Áreas de la Organización
- ❖ Plantilla DCS - Arquitectura de Información
- ❖ Plantilla DCS - Informe del Levantamiento de Información para la Arquitectura de Información v1.0
- ❖ Plantilla DCS – Presupuesto

1.4 Conclusiones del Capítulo

En este capítulo se realizó un estudio de las metodologías estructuradas, teniendo en cuenta su definición, clasificación, características de las mismas y la situación actual con el objetivo de identificar una que fuera factible en el control y organización de la documentación generada por los proyectos basados en

programación estructurada en el Área Temática Herramientas para la Teleformación, obteniéndose como propuesta Métrica V3, de la misma se analizó objetivos, características, aportaciones principales y describiendo brevemente sus procesos e interfaces. Además se llevó a cabo un análisis del Expediente de Proyecto y de los Lineamientos de Calidad de Software definidos en la UCI.

CAPITULO 2: PROPUESTA DEL EXPEDIENTE DE PROYECTO

Introducción

En el presente capítulo se realiza la formulación de un Expediente de Proyecto para los proyectos de desarrollo, servicio de personalización e instalación, con el objetivo de regir el proceso de desarrollo de software de los proyectos del Área Temática Herramientas para la Teleformación perteneciente al Polo Software Educativo que utilizan programación estructurada, teniendo en cuenta el estudio realizado en el capítulo anterior. Además, se realiza un análisis profundo acerca de las fases e interfaces de Métrica V3, así como también una especificación detallada de los requerimientos de cada planilla.

2. Surgimiento del Expediente de Proyecto basado en Métrica 3

La formulación del Expediente de Proyecto basado en Métrica V3 surge como una necesidad primeramente de los líderes de módulos, que mediante la implementación de los mismos, descubrieron que la metodología de desarrollo de software propuesta por la universidad no era aplicable a la programación utilizada, y es ahí donde surge la iniciativa de proponer otra metodología basada en programación estructurada, para así organizar y perfeccionar el proceso de desarrollo de software existente en los módulos de Teleformación. Esta propuesta metodológica también requería una guía organizativa que satisficiera el proceso metodológico y controlara la documentación generada.

La propuesta de expediente está conformada por dos grupos principales, procesos e interfaces, los cuales hacen posible que la metodología antes mencionada organice y planifique los sistemas de información.

[18]:

- ❖ Procesos:
 - ✓ Planificación del Sistema de Información
 - ✓ Desarrollo del Sistema de Información
 - ✓ Mantenimiento del Sistema de Información
- ❖ Interfaces

- ✓ Gestión de Proyectos
- ✓ Seguridad
- ✓ Gestión de Configuración
- ✓ Calidad

Los grupos que conforman el Expediente de Proyecto contienen a su vez varios subgrupos los cuales recogen gran parte de la información almacenada en el expediente propuesto por la UCI, además de información particular de la metodología Métrica V3.

Planificación del Sistema de Información

El grupo Planificación de Sistemas de Información tiene como objetivo la obtención de un marco de referencia para el desarrollo de sistemas de información que responda a los objetivos estratégicos de la organización. Este grupo contiene toda la información referente a la situación actual y futura del proyecto, reflejándola en el Catálogo de Requisitos del PSI, así como también la Arquitectura de Información del mismo. La perspectiva del plan debe ser estratégica y operativa, no tecnológica. Por otra parte se evalúan las opciones tecnológicas y se propone un entorno [18].

Desarrollo del Sistema de Información

El grupo Desarrollo del Sistema de Información tiene como objetivo lograr el desarrollo completo de los sistemas de información. El mismo contiene varios subprocesos que son lo que se encargan de lograr y dar valor al objetivo, Estudio de la Viabilidad del Sistema, Análisis del Sistema de Información, Diseño del Sistema de Información, Construcción del Sistema de Información e Implantación del Sistema de Información. Todos ellos recogen y almacenan la información relacionada con el estudio del proceso en todas sus dimensiones, es decir, la selección de los requisitos, modelos propuestos para solución, casos de usos, diseño, arquitectura, pruebas e implantación. Este proceso es el de mayor relevancia en Métrica V3, ya que describe, casi todo el proceso de desarrollo de software de la misma.

Mantenimiento del Sistema de Información

El objetivo de este grupo es la obtención de un sistema de información desarrollado con MÉTRICA Versión 3, a partir de las peticiones de mantenimiento que los usuarios realizan con motivo de un problema detectado en el sistema, o por la necesidad de una mejora del mismo. Este grupo tiene gran relación con la Gestión de Configuración, y el control de los cambios realizados en la misma. Dicho grupo recomienda, llevar un catálogo de peticiones de mantenimiento sobre los sistemas de información, en el que se registren una serie de datos que permitan disponer de la información antes mencionada.

Gestión de proyecto

La Gestión de Proyectos tiene como finalidad principal la planificación, el seguimiento y control de las actividades y de los recursos humanos y materiales que intervienen en el desarrollo de un Sistema de Información. Como consecuencia de este control es posible conocer en todo momento qué problemas se producen y resolverlos o paliarlos de manera inmediata. Este grupo recoge los documentos necesarios para la administración efectiva de un proyecto software. Se traduce en planes para la ejecución de cada uno de los procesos presentes en el desarrollo de software así como aquellos aspectos generales del proyecto (cronogramas, recursos, presupuestos, seguridad, cuaderno de trabajo, planificación...).

Aseguramiento de la Calidad

El grupo Aseguramiento de la Calidad proporciona un marco común de referencia para la definición y puesta en marcha de planes específicos de aseguramiento de calidad aplicables a proyectos concretos. Para lograr la calidad de los productos, en este grupo se realiza la plantilla Plan de Aseguramiento de la Calidad el cual se aplicará durante la planificación del proyecto de acuerdo a la estrategia de desarrollo adoptada en la gestión del proyecto. En el Plan de Aseguramiento de la Calidad se reflejan las actividades de calidad a realizar (normales o extraordinarias), los estándares a aplicar, los productos a revisar, los procedimientos a seguir en la obtención de los distintos productos durante el desarrollo en MÉTRICA Versión 3 y la normativa para informar de los defectos detectados a sus responsables y realizar el

seguimiento de los mismos hasta su corrección. Además, se realiza un Glosario de Términos, Listas de Chequeo, Plan de Mediciones, Plan de Pruebas y la recogida de las no conformidades presentadas en el proyecto, dichas plantillas también contribuyen al aseguramiento de la calidad [18].

Gestión de Configuración

El objetivo del grupo gestión de la configuración es mantener la integridad de los productos que se obtienen a lo largo del desarrollo de los sistemas de información, garantizando que no se realizan cambios incontrolados y que todos los participantes en el desarrollo del sistema disponen de la versión adecuada de los productos que manejan.

En este grupo se realiza un Plan de Gestión de Configuración con el objetivo de controlar los elementos de configuración software, las actividades de la Gestión de Configuración, entre otros aspectos que son de vital importancia en el proceso de control de cambios. La gestión de configuración como se mencionó en el grupo de Mantenimiento de Sistemas de Información facilita el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo, lo que se traduce en un aumento de calidad del producto, de la satisfacción del cliente y, en consecuencia, de mejora de la organización.

Seguridad

El objetivo del grupo seguridad es incorporar en los sistemas de información mecanismos de seguridad adicionales a los que se proponen en la propia metodología, asegurando el desarrollo de cualquier tipo de sistema a lo largo de los procesos que se realicen para su obtención, para ello se realiza un Plan de Seguridad en el cual se recoge políticas de seguridad, control de actividades para preservar la seguridad en el proyecto y entorno de trabajo, objetivos de seguridad así como las funciones de la misma.

2.1 Estructura del Expediente de Proyecto basado en Métrica 3

En la confección de la propuesta de Expediente de Proyecto se definieron plantillas que permitirán estandarizar la documentación de los módulos del Área Temática Herramientas para la Teleformación. La elaboración de las plantillas se realizó a partir de la revisión de los procesos e interfaces de Métrica V3, y del estudio de los Lineamientos de Calidad de Software propuestos por la UCI.

El Expediente de Proyecto basado en Métrica V3 quedó conformado para proyectos de Desarrollo de Software, de Servicio y de Instalación. Este expediente contiene dos grupos **Procesos** e **Interfaces**, a continuación se muestran las plantillas que recoge cada grupo:

2.1.1 Expediente de Proyecto para Proyectos de Desarrollo

Procesos

❖ **Planificación del Sistema de Información**

Plantilla DCS- Catálogo de Requisitos del PSI v1.0

Plantilla DCS- Arquitectura de Información v1.0

❖ **Desarrollo del Sistema de Información**

✓ **Estudio de la Viabilidad del Sistema**

Plantilla DCS- Alternativas de Solución v1.0

Plantilla DCS- Documento Visión v1.0

✓ **Análisis del Sistema de Información**

Plantilla DCS- Plan de Gestión de Requisitos v1.0

Plantilla DCS- Plan de Migración y Carga Inicial de Datos v1.0

✓ **Diseño del Sistema de Información**

Plantilla DCS- Documento de Arquitectura del Sistema v1.0

- ✓ **Construcción del Sistema de Información**

Plantilla DCS- Entorno de Construcción v1.0

- ✓ **Implantación y Aceptación del Sistemas**

Plantilla DCS- Plan de Implantación v1.0

- ❖ **Mantenimiento del Sistema de Información**

Plantilla DCS- Plan de Mantenimiento v1.0

Interfaces

- ❖ **Gestión de Proyectos**

- ✓ **Plan de proyectos**

Plantilla DCS- Cuaderno de Trabajo del Ingeniero v1.0

Plantilla DCS- Planificación General del Proyecto v1.0

Plantilla DCS- Estimación de Esfuerzo v1.0

Plantilla DCS- Glosario de Términos v1.0

- ✓ **Recursos**

Plantilla DCS- Entorno Tecnológico v1.0

Plantilla DCS- Plan de Capacitación v1.0

Plantilla DCS- Registro de Recursos v1.0

Plantilla DCS- Roles y Responsabilidades v1.0

- ✓ **Reuniones**

Plantilla DCS- Minuta de Reuniones v1.0

- ✓ **Revisiones**

Plantilla DCS- Plan de Revisiones v1.0

- ✓ **Riesgos**

Plantilla DCS- Plan de Mitigación de Riesgos v1.0

❖ **Aseguramiento de la Calidad**

Plantilla DCS- Plan de Aseguramiento de la Calidad v1.0

Plantilla DCS- Listas de Chequeo v1.0

Plantilla DCS- No Conformidades (Ampliada) v1.0

Plantilla DCS- Plan de Mediciones v1.0

Plantilla DCS- Respuesta a No Conformidades v1.0

Plantilla DCS- Plan de Pruebas v1.0

❖ **Gestión de Configuración**

Plantilla DCS- Plan de Gestión de Configuración v1.0

Plantilla DCS- Formulario de Pedido de Cambio v1.0

Plantilla DCS- Solicitud de Cambio v1.0

❖ **Seguridad**

Plantilla DCS- Plan de Seguridad v1.0

❖ **Legal**

Plantilla ALBET Acta de Entrega

Plantilla ALBET Acta de inicio de proyecto

Plantilla ALBET Acta de Terminación de Proyecto

Plantilla ALBET – Carta

Plantilla ALBET Informe de Resultado

Plantilla ALBET Minutas de Reunión

Plantilla ALBET – Principales Indefiniciones

Plantilla ALBET – Proyecto Técnico

2.1.2 Expediente de Proyecto para Proyectos de Servicios de Personalización

Procesos

❖ **Planificación del Sistema de Información**

Plantilla DCS- Arquitectura de Información v1.0

- ✓ **Desarrollo del Sistema de Información**
- ✓ **Estudio de la Viabilidad del Sistema**

Plantilla DCS- Documento Visión v1.0

- ✓ **Análisis del Sistema de Información**
- ✓ **Diseño del Sistema de Información**

Plantilla DCS- Documento de Arquitectura del Sistema v1.0

- ✓ **Construcción del Sistema de Información**

Plantilla DCS- Entorno de Construcción v1.0

- ✓ **Implantación y Aceptación del Sistemas**

Plantilla DCS- Plan de Implantación v1.0

❖ **Mantenimiento del Sistema de Información**

Plantilla DCS- Plan de Mantenimiento v1.0

Interfaces

❖ **Gestión de Proyectos**

- ✓ **Plan de proyectos**

Plantilla DCS- Cuaderno de Trabajo del Ingeniero v1.0

Plantilla DCS- Planificación General del Proyecto v1.0

Plantilla DCS- Estimación de Esfuerzo v1.0

Plantilla DCS- Glosario de Términos v1.0

- ✓ **Recursos**

Plantilla DCS- Entorno Tecnológico v1.0

Plantilla DCS- Plan de Capacitación v1.0

Plantilla DCS- Registro de Recursos v1.0

Plantilla DCS- Roles y Responsabilidades v1.0

✓ **Reuniones**

Plantilla DCS- Minuta de Reuniones v1.0

✓ **Revisiones**

Plantilla DCS- Plan de revisiones v1.0

✓ **Riesgos**

Plantilla DCS- Plan de Mitigación de Riesgos v1.0

❖ **Aseguramiento de la Calidad**

Plantilla DCS- Plan de Aseguramiento de la Calidad v1.0

Plantilla DCS- No Conformidades (Ampliada) v1.0

Plantilla DCS- Plan de Mediciones v1.0

Plantilla DCS- Respuesta a No Conformidades v1.0

❖ **Gestión de Configuración**

Plantilla DCS- Plan de Gestión de Configuración v1.0

Plantilla DCS- Formulario de Pedido de Cambio v1.0

❖ **Seguridad**

Plantilla DCS- Plan de Seguridad v1.0

❖ **Legal**

Plantilla ALBET Acta de Entrega

Plantilla ALBET Acta de inicio de proyecto

Plantilla ALBET Acta de Terminación de Proyecto

Plantilla ALBET – Carta

Plantilla ALBET Informe de Resultado

Plantilla ALBET Minutas de Reunión
Plantilla ALBET – Principales Indefiniciones
Plantilla ALBET – Proyecto Técnico

2.1.3 Expediente de Proyecto para Proyectos de Instalación

❖ Legal

Plantilla ALBET Acta de Entrega
Plantilla ALBET Acta de inicio de proyecto
Plantilla ALBET Acta de Terminación de Proyecto
Plantilla ALBET – Carta
Plantilla ALBET Informe de Resultado
Plantilla ALBET Minuta de Reuniones
Plantilla ALBET – Principales Indefiniciones
Plantilla ALBET – Proyecto Técnico

❖ Instalación

Plantilla DCS – Plan de Implantación
Plantilla DCS – Plan de Pruebas
Plantilla DCS – Diagrama de Despliegue
Plantilla DCS – Cronograma
Plantilla DCS – Documento Visión
Plantilla DCS – Lista de Riesgos
Plantilla DCS – Solicitud
Plantilla DCS – Requisitos Necesarios para la Instalación
Plantilla DCS – Minutas de Reunión
Plantilla DCS – Roles y Responsabilidades

2.2 Métodos de Estimación de Métrica V3

Métrica V3 plantea en la interfaz Gestión de Proyecto, dos métodos de estimación para el cálculo del esfuerzo realizados por los integrantes del equipo de trabajo, Albrecht y Mark II.

2.2.1 Método Albrecht

El método Albrecht fue propuesto por Alan Albrecht, colaborador de International Business Machines (IBM), de ahí el nombre que lleva, esta propuesta partía de un concepto distinto, el desarrollo de un índice de la funcionalidad de un sistema como una medida asociada a su tamaño. Desde entonces los métodos han perfeccionado basándose en la experiencia acumulada. El proceso de medición se realiza en el contexto de un modelo abstracto del sistema. Este modelo se compone de transacciones y archivos, estos elementos se identifican a partir de los documentos de requerimientos, técnicas de diseño estructurado u otros modelos.

Cuando Albrecht desarrolló el Análisis de Puntos de Función (FPA) tenía por objetivos [19]:

- ❖ Un método significativo para el usuario final. Al estar relacionados directamente a los requerimientos del usuario. Reglas para contar los Puntos Función (FP) que deberían ser fáciles de aplicar.
- ❖ FP que deberían ser estimables a partir de la especificación de requerimientos.
- ❖ FP que deberían ser independientes de la tecnología usada para desarrollar el sistema.

2.2.1.1 Principales componentes del método Albrecht

A continuación se presentan los componentes del Método de Puntos Función de Albrecht:

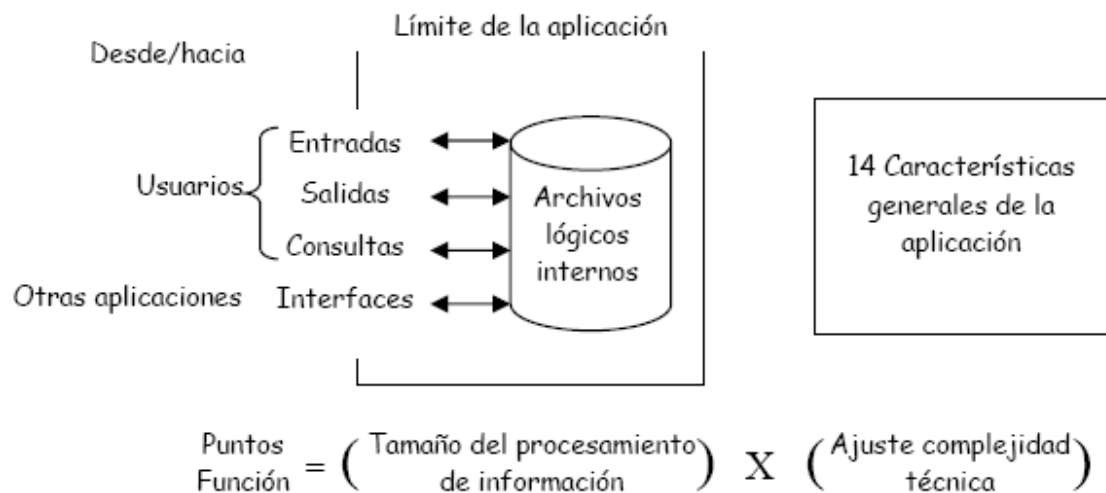


Figura 2. Componentes del Método de Puntos Función de Albrecht [19]

Según Albrecht debido a que los FP miden los sistemas desde una perspectiva funcional, son independientes de la tecnología. Independientemente del lenguaje, método de desarrollo o plataforma de hardware usados, la cantidad de FP permanecerá constante. La única variable es la cantidad de esfuerzo necesario para entregar un conjunto de FP. Por lo tanto, el FPA se puede usar para determinar si una herramienta, un entorno o un lenguaje es más productivo comparado con otros, dentro de una organización o entre organizaciones.

Entradas Externas (EI): es un proceso elemental en el cual los datos cruzan el límite desde el exterior hacia el interior del sistema. Estos datos pueden venir desde una pantalla de entrada de datos o desde otra aplicación. Los datos pueden ser usados para mantener uno o más archivos lógicos internos. Ver figura 3 [19].

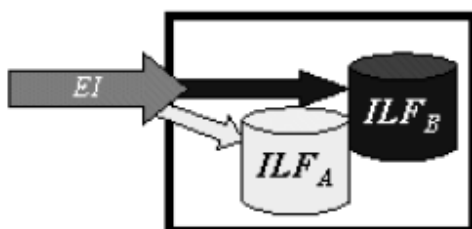


Figura 3. Ficheros Lógicos Internos (ILF) [19]

Salidas externas (EO): es un proceso elemental en el cual los datos derivados atraviesan el límite desde el interior hacia el exterior del sistema. Un EO puede actualizar un archivo lógico interno. Los datos crean reportes o archivos de salida hacia otras aplicaciones. Esos reportes y archivos son creados a partir de uno o más archivos lógicos internos y archivos de interfaz externa. Ver figura 4 [19].

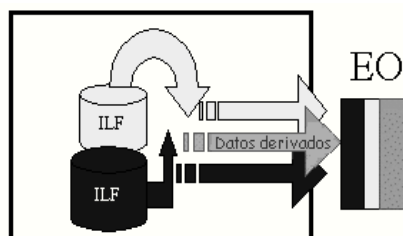


Figura 4. EO que referencia y deriva información [19]

Consultas Externas (EQ): Es un proceso elemental con componentes de entrada y salida que resulta en el entorno de datos desde uno o más archivos lógicos internos y archivos de interfaces externas. El proceso de entrada no actualiza archivos y la salida no contiene datos derivados [19].

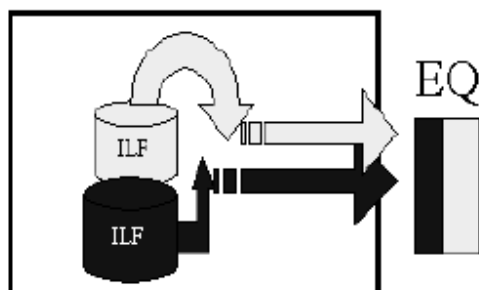


Figura 5. EQ con dos ILF y sin datos derivados [19]

Una vez que todos los componentes han sido clasificados como EI, EO, EQ, ILF o EIF, se le asigna una posición en la escala que puede ser: bajo, medio o alto. Para las transacciones (EI, EO, EQ) esa posición se basa en la cantidad de archivos actualizados o referenciados (FTR) y la cantidad de tipos de elementos dados. Los tipos de elementos de datos (DET), son un campo único, no recursivo, reconocido por el usuario, en el caso particular de las EQ se les asigna un factor como EI, si el mismo FTR se usa en la misma entrada y salida, se cuenta solo una vez, si el mismo DET se usa en la parte de entrada y salida, también se cuenta una sola vez [19].

Cada una de las siguientes tablas se utiliza en el proceso de asignación de posiciones en la escala y sus factores asociados [19]:

FTR	Elementos Dato		
	1-4	5-15	>15
0-1	Bajo	Bajo	Medio
2	Bajo	Medio	Alto
3 o más	Medio	Alto	Alto

Tabla 2. Entradas Externas (EI)

FTR	Elementos Dato		
	1-5	6-19	>19
0-1	Bajo	Bajo	Medio
2-3	Bajo	Medio	Alto
>3	Medio	Alto	Alto

Tabla 3. EO y EQ

Posición			Factor
	EO	EQ	EI
Bajo	4	3	3
Medio	5	4	4
Alto	7	6	6

Tabla 4. Transacciones

Para los archivos ILF y Ficheros Lógicos Externos (EIF) la posición en la escala (bajo, medio, alto) se basa en los tipos de elementos registro (RET) y los DET. Un RET es un subgrupo de datos reconocidos por el usuario dentro de un ILF o EIF [19].

RET	Elementos Dato		
	1-19	20-50	>50
1	Bajo	Bajo	Medio
2-5	Bajo	Medio	Alto
>5	Medio	Alto	Alto

Tabla 5. Tipo Registro

Posición	Factor	
	ILF	EIF
Bajo	7	5
Medio	10	7
Alto	15	10

Tabla 6. ILF y EIF

Las cuentas para cada nivel de complejidad de cada tipo de componente se pueden ingresar en una tabla como la siguiente (Tabla 7). Cada cuenta se multiplica por el puntaje numérico indicado para determinar el puntaje total. Los valores de cada fila se suman a través de la tabla resultando el valor total para cada tipo de componente [19].

Tipo de Componente	Complejidad de los componentes			
	Bajo	Medio	Alto	Total
Entradas Externas (EI)	*3 =	*4 =	*6 =	
Entradas Internas (EO)	*4 =	*5 =	*7 =	
Consultas Externas (EQ)	*3 =	*4 =	*7 =	
Archivos Lógicos Internos (ILF)	*7 =	*10 =	*15 =	
Archivos de Interface Externa (EIF)	*5 =	*7 =	*10 =	
	UFP			
	TCA			
	FP			

Tabla 7. Cálculo de Puntos Función de Albrecht

El valor de Ajuste de Complejidad Técnica (TCA) se obtiene a partir de las 14 características generales del sistema que asignan un puntaje a la funcionalidad general de la aplicación que se está midiendo. Esas características tienen descripciones asociadas que ayudan a determinar los grados de influencia (DI) de las características. Los grados de influencia varían en una escala de cero a cinco desde ninguna a fuerte influencia. El objetivo de este factor es considerar la influencia sobre el tamaño funcional de los requerimientos de calidad y técnicos, en la tabla 8 se detallan las características generales [19].

Características Generales del Sistema			
1	Comunión de datos	8	Actualización interactiva
2	Procesamiento de datos distribuidos	9	Complejidad de procesamiento
3	Performance	10	Reusabilidad
4	Entorno Operativo muy utilizado	11	Facilidad de instalación
5	Frecuencia de Transacción	12	Facilidad de operación
6	Entrada de datos Interactiva	13	Múltiples Instalaciones
7	Eficiencia usuario final	14	Facilidad de Cambios

Tabla 8. Características Generales del Sistema por Albrecht FPA

Una vez que han sido valorados los 14 factores, debe aplicarse la ecuación de Ajuste de complejidad técnica (TCA) [19].

$$TCA = 0.65 + 0.01 * \sum DI$$

Donde DI son los grados de influencia dados por cada una de las 14 características generales del sistema. La cuenta final de FP se obtiene multiplicando el valor de la cantidad total de FP sin ajustar (UFP) por TCA. Se completa la Tabla 8 con el valor TCA y se obtiene FP [19].

$$FP = UFP * TCA$$

2.2.1.2 Críticas del Método Albrecht

A pesar de ser una idea muy interesante y novedosa, al experimentar con el uso del método de Albrecht surgieron una cantidad de dificultades y anomalías. Entre ellas se puede mencionar [20]:

- ❖ Los componentes propuestos por Albrecht a veces son difíciles de identificar en la práctica del desarrollo de sistemas.
- ❖ Es razonable preguntarse como determinó Albrecht los pesos (o puntajes) usados para los diferentes tipos de componentes y para los diferentes niveles de complejidad cuando se calculan los UFP.
- ❖ Una crítica similar se puede hacer respecto a la elección de los componentes y los pesos para las características generales de la aplicación del Ajuste de la Complejidad Ciclomática.
- ❖ Otra cuestión que se plantea es si este método trata adecuadamente con la complejidad de procesamiento interno del sistema.
- ❖ Por último, las reglas de Albrecht parecen favorecer a los sistemas discretos frente a los sistemas integrados.

Las más serias deficiencias del Método de Albrecht son aquéllas que surgen a partir del estrecho rango de puntajes en FP que se puede adjudicar a transacciones con un rango muy amplio de variación en la cantidad de DET en la entrada/salida y en la complejidad de procesamiento interno.

2.2.2 Método Mark II

A partir del método Albrecht y de las críticas que se han expuesto anteriormente, Charles Symons líder del proyecto conjunto The Common Software Measurement International Consortium (COSMIC) desarrolló a fines de los '80 un nuevo enfoque basándose en el mismo concepto. Este método se conoce como Análisis de Puntos Función Mark II o simplemente MKII FPA.

Este método puede usarse para medir el tamaño funcional de cualquier aplicación de software que se pueda describir en términos de transacciones lógicas, cada una comprendiendo componentes de entrada, proceso y salida. Las reglas fueron diseñadas para aplicarlas a sistemas de información de gestión donde

el componente procesamiento de cada transacción tiende a estar dominado por consideraciones de almacenamiento y recuperación de datos.

El mismo puede ser aplicable a otros dominios de software, pero debe notarse que las reglas no toman en cuenta las contribuciones al tamaño de algoritmos complejos como aquéllos que se encuentran en el software científico y de ingeniería, ni toman en cuenta los requerimientos de tiempo real [21].

2.2.2.1 Componentes del Método Mark II

En la Figura 21 se describen los componentes del Método MKII. Los cambios más significativos respecto al método analizado anteriormente están relacionados al tamaño del componente procesamiento de información. La idea es considerar al sistema como una colección de transacciones lógicas, para ello es necesario determinar previamente el límite del sistema, a fin de establecer cuáles pertenecen al sistema que se medirá y cuáles quedarán excluidos. La aplicación a parte de la aplicación incluida dentro del límite debe ser un cuerpo coherente de funcionalidad que comprende una o más transacciones lógicas [20].

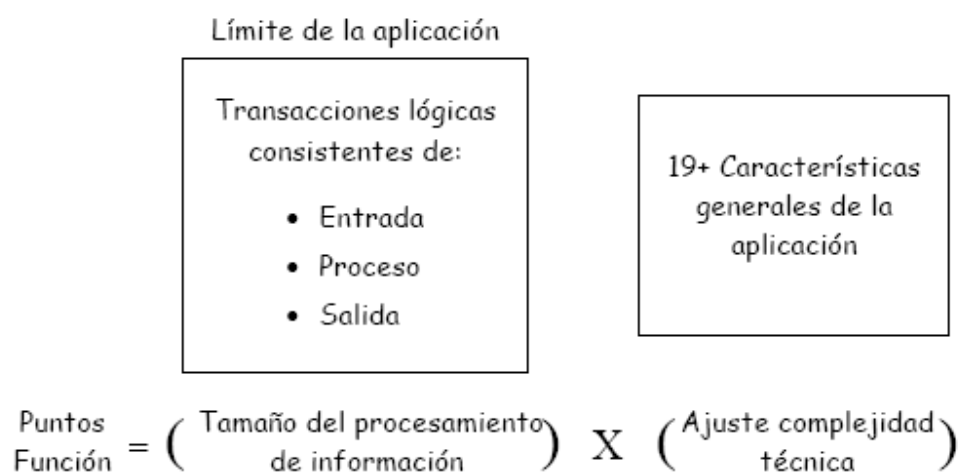


Figura 6. Componentes del Método de Puntos Función Mark II [20]

Una transacción lógica se define como una combinación de tres componentes [21]:

- ❖ Un elemento de entrada a través del límite.
- ❖ Un elemento de procesamiento de los datos dentro del límite.
- ❖ Un elemento de salida a través del límite.

2.2.2.2 Componente de Procesamiento en las Transacciones Lógicas

El tamaño de este componente es proporcional a la cantidad de entidades de datos referenciados en cada transacción, entendiéndose como referencia a los procesos de creación, lectura, actualización, eliminación y listado. Se encuentra la cantidad de tipos de entidades primarias que son referenciadas por la transacción lógica más la referencia a una entidad denominada System.

El concepto de entidad se define como: objeto, transacción, período de tiempo, tangible o intangible acerca de la cual se requiere mantener información. Una entidad es un elemento cuyas ocurrencias son identificables en forma individual o colectiva [20].

Dentro del límite de la aplicación puede haber solo una entidad System y como máximo se puede incluir una referencia en el componente procesamiento de una transacción lógica.

Este método asume que el tamaño relativo de los componentes entrada y salida en cada transacción lógica es proporcional a la cantidad de DET en cada componente respectivo. Un DET es un ítem único de información que es indivisible para el propósito de la transacción lógica y es parte de un flujo de datos en la entrada y salida a través del límite [21].

2.2.2.3 Pasos para calcular el Índice de Puntos Función

Para calcular el Índice de Puntos Función (FPI), se deben seguir los siguientes pasos [20]:

- ❖ Obtener una comprensión general del sistema al cual se aplicará el método, para lograr el conocimiento del problema y del sistema necesario para cumplir con los requerimientos del usuario.
- ❖ Definir el punto de vista, propósito de la medición y límite del sistema que se medirá, para establecer qué elementos estarán incluidos dentro del mismo.
- ❖ Determinar las entidades primarias (representan elementos principales para el sistema en estudio) y no primarias (tienen pocos atributos y generalmente tienen relación con tablas maestras que sirven para propósitos de validación). (Ver tabla 9).

Criterio	Entidad Primaria	Entidad no Primaria
Numero de Atributos	Varios con valores que cambias frecuentemente	Muy pocos, los valores cambian raramente
Frecuencia de Ocurrencia	Pueden cambiar a menudo	Fijo permanentemente o cambia raramente
Tipo de cambio de los valores atributos	Durante la operación normal del sistema	Usualmente fuera de la operación normal del sistema
Autorización de cambios de los valores atributos	Ninguna especial, ejecutados por el usuario normal del sistema	Ejecutados por el administrador del sistema
Ciclo de vida de la entidad	Usualmente varias etapas o estados	Usualmente solo "vive" o no existente

Tabla 9. Criterios para distinguir entidades primarias [20]

- ❖ Determinar las transacciones lógicas.
Siempre hay que tener presente: ¿Qué quiere el Usuario?
Hay algunas reglas que permiten diferenciar algunos procesos para determinar si se consideran o no transacciones [20].

- ✓ Los procesos internos que se repiten con frecuencia deben ser contados en cada transacción que ocurren. No deben contarse como transacciones lógicas separadas a menos que sean disparados en forma independiente por un evento significativo para el usuario.
 - ✓ Los cambios en el valor de un DET de entrada no pueden disparar transacciones lógicas separadas.
 - ✓ Los eventos que son significativos para el diseñador físico del sistema no se cuentan como eventos que disparan transacciones lógicas.
- ❖ Determinar los componentes de las transacciones lógicas. Cada una de ellas consta de tres componentes: entrada, procesamiento y salida.
- ✓ El componente de entrada a través del límite consiste de la adquisición y validación de los datos ingresados que describen un evento de interés en el mundo externo o de los parámetros de una consulta para obtener información de salida desde la aplicación.
 - ✓ El componente de procesamiento consiste en el almacenamiento y recuperación de información que describe el estado de las entidades de interés en el mundo externo.
 - ✓ El componente de salida a través del límite del formateo y presentación de información hacia el mundo externo.

Para organizar el trabajo este método utiliza una tabla de transacciones lógicas con el siguiente encabezado:

ID	Nombre transacción	Evento o consulta	No de DET entrada	Respuesta	No de DET salida	Tipo de entidad referenciada	No de ER	MKII FP
----	--------------------	-------------------	-------------------	-----------	------------------	------------------------------	----------	---------

Tabla 10. Encabezado de la tabla Transacciones Lógicas

- ✓ Aplicar la fórmula para calcular FPI.

El método Mark II utiliza la siguiente fórmula para determinar el Índice de Puntos Función o Tamaño Funcional [21]:

$$FPI = W_I * \sum N_I + W_E * \sum N_E + W_O * \sum N_O$$

Donde W_i , W_e y W_o representan los pesos promedios en la industria para los componentes entrada (I), entidades referencias (E) y de salida (O). Sus valores son:

$W_i = 0.58$, $W_e = 1.66$, $W_o = 0.26$ y $N_i =$ Cantidad de DET de entrada, $N_e =$ Cantidad de entidades referenciadas y $N_o =$ Cantidad de DET de salida.

Los pesos promedios en la industria suman 2.5 para mantener la correspondencia con los FP de Albrecht. En promedio los métodos dan aproximadamente los mismos tamaños, hasta cerca de los 400 FP. Para sistemas más grandes Mark II tiende a producir tamaños mayores que el método de Albrecht.

- ✓ Opcional: En caso de necesitarse calcular los FP considerando los requerimientos técnicos se debe calcular el TCA.

Para el cálculo del factor TCA, se utiliza una lista de características generales de la aplicación que se basa en la que usa Albrecht, pero se extiende a 19 características, o más si realmente se justifica. Las características adicionales de Mark II son:

Características Generales del Sistema			
15	Requerimientos de otras aplicaciones	18	Uso directo por terceras partes
16	Seguridad, privacidad, auditabilidad	19	Documentación
17	Necesidad de entrenamiento al usuario	20	Características definidas por el usuario

Tabla 11. Características Generales del Sistema adicionales para MK II [20]

$$TCA = (TDI * C) + 0.65 [21]$$

Donde el valor actual promedio en la industria de C es 0.005 y el TDI es el total de los puntajes para cada una de las 19 características, llamado grado de influencia total. Los grados de influencia al igual que en el método de Albrecht varían en una escala de 0 a 5, desde ninguna a fuerte influencia. [20] Debe notarse que el valor propuesto para el factor C en el cálculo del TCA, es la mitad del propuesto por Albrecht. La interpretación es que estos factores técnicos actualmente se obtienen con menor esfuerzo que muchos años atrás. El TCA no está incluido dentro de la norma ISO/IEC 14143 y generalmente no se recomienda su uso [21].

Luego el Índice de Puntos Función Ajustado (AFPI), se expresa como [21]:

$$\text{AFPI} = \text{FPI} * \text{TCA}$$

2.2.3 Comparación entre los Métodos Albrecht y Mark II

Este análisis se realiza con el objetivo de mostrar las similitudes y diferencias entre las dos variantes de FPA estudiadas en el epígrafe anterior. Las técnicas de FPA intentan cuantificar los requerimientos funcionales y no funcionales. En ambas técnicas esto se logra mediante la determinación inicial del tamaño funcional y la posterior aplicación de un factor de ajuste al tamaño, para atender el esfuerzo adicional necesario para cumplir los requerimientos de calidad. Las principales diferencias entre Albrecht Y Mark II se encuentran en la forma en que se determina el tamaño funcional.

En cualquier técnica de medición del tamaño funcional se miden los requerimientos funcionales de una aplicación en términos de uno o más tipos de componentes lógicos, cada uno de los cuales debe tener una relación explícita con el límite de la aplicación. Además, se proveen reglas para identificar los tipos de componentes lógicos del software y asignarles un valor numérico que representa su contribución al tamaño funcional. Después de identificar los componentes, se determina el tamaño funcional de la aplicación mediante la suma del tamaño de cada uno de los tipos de componentes lógicos [21].

Mark II y Albrecht expresan los requerimientos funcionales en términos de componentes lógicos. Mark II usa un único tipo de componente lógico y expresa todos los requerimientos funcionales como transacciones lógicas. Albrecht usa cinco tipos de componentes lógicos: EI, EO, EQ, ILF y EIF [21].

Los tipos de componentes lógicos están formados por varias partes. En Mark II cada transacción lógica se compone de entrada, procesamiento y salida. En Albrecht, EI y EO se componen de mensajes de entrada y salida respectivamente, EQ por un par entrada/salida, ILF y EIF por datos persistentes mantenidos por la aplicación u otra aplicación, respectivamente.

En Mark II, cada tipo de entidad se trata de forma independiente y se cuentan las referencias a las entidades por cada transacción lógica. En Albrecht, los tipos de entidad se agrupan para formar ILF si están dentro del límite de la aplicación o EIF si están fuera del límite. Las referencias a los tipos de entidad se cuentan como FTR por cada EI, EO o EQ [21].

En ambas técnicas hay una relación entre los tipos de componentes lógicos y el límite de aplicación.

En Mark II, por cada transacción lógica, el mensaje de entrada y de salida atraviesa el límite cuando llega y sale de la aplicación respectivamente y la parte de proceso está completamente retenida dentro del límite. En Albrecht, cada EI o EO atraviesa el límite cuando llega y sale de la aplicación, por cada EQ la parte de entrada y salida atraviesa el límite cuando llega y sale, los ILF y EIF están completamente retenidos dentro del límite de la aplicación que se analiza o de otra aplicación, respectivamente.

Ambas técnicas basan sus reglas para evaluar el tamaño de sus componentes lógicos en las cuentas bases. En ambas se cuenta la cantidad de DET en las partes del mensaje del componente lógico y la cantidad de referencias a datos persistentes dentro de la aplicación, aunque usan terminología, las definiciones son similares [21].

Mientras en Mark II las cuentas resultan de las entradas y salidas de las transacciones y de las referencias a los datos persistentes, en Albrecht se identifican ILF y EIF y se cuentan sus partes constituyentes que contribuyen a la funcionalidad [21].

En Mark II se cuenta una referencia a una entidad por cada tipo de entidad accedida durante el curso de una transacción lógica, mientras que en Albrecht se cuenta una FTR por cada ILF o EIF accedido durante el curso de entrada, salida o consulta.

Los objetos de la especificación que se deben identificar son los mismos para ambas técnicas: mensajes de entrada y salida, mensajes de error y tipos de entidad. En Albrecht se consideran los DET almacenados en los tipos de entidad.

Las principales diferencias entre las dos técnicas surgen desde cómo se construyen las cuentas bases y no desde qué es lo que se cuenta.

Para determinar las cuentas bases, se identifican los respectivos componentes lógicos y se cuenta la cantidad de DET en los mensajes asociados. Los mensajes de error se tratan de manera diferente en las dos técnicas. En Albrecht los atributos de mensaje de error se tratan como atributos de un mensaje de entrada, en Mark II se tratan como atributos adicionales de un mensaje de salida. A continuación se cuenta la cantidad de accesos a datos persistentes. En Albrecht, además, se deben ejecutar pasos adicionales para contar la cantidad de tipos de entidad en los grupos de tipos de entidad que forman cada ILF y EIF y contar la cantidad de DET almacenados en los tipos de entidad [21].

Hay algunas otras variaciones. Mark II usa el concepto de entidad System para agrupar las entidades no primarias que pueden contener información dependiente de la implementación. Albrecht no cuenta datos dependientes de la implementación.

En Mark II las cuentas base se usan directamente en los cálculos del tamaño funcional expresado en FP sin ajustar. En Albrecht las cuentas base se usan para determinar la magnitud de cada componente lógico.

Usando las tablas provistas por el método se valora la magnitud de cada componente como bajo, medio o alto, en base a los valores de las cuentas base de DET y referencias a tipos de entidad o la cantidad de RET en el caso de ILF y EIF. Albrecht usa el término complejidad en vez de magnitud, no debe confundirse con la complejidad técnica [21].

Los mensajes de entrada, salida y referencias a datos persistentes tienen su propia contribución a una aplicación, pero la contribución de cada una es de diferente clase. Para combinarlas y derivar un único valor numérico del Índice de Tamaño Funcional, se deben normalizar las cuentas base para usar una única unidad. Esto se logra a través de los pesos relativos.

En Mark II se usan tres pesos: W_i , W_o y W_e . En Albrecht, el sistema de pesos es más complicado, debido a la gran cantidad de tipos de componentes lógicos. Cada tipo de componente lógico tiene asignado un peso que depende del tipo de componente y su magnitud [21].

En ambas técnicas el Índice de Tamaño Funcional de la aplicación expresado en FP se obtiene a partir de la suma de las cuentas afectadas por los pesos.

Los dos métodos usan enfoques similares para considerar los requerimientos técnicos. Se evalúa una lista de características no funcionales en una escala de cero a cinco. Albrecht 14 características, Mark II usa las mismas 14 pero agrega otras cinco o más.

Una vez evaluado el grado de influencia de las características, se suman para dar el grado de influencia total. Este valor se usa para ajustar el Índice de Tamaño Funcional y obtener un nuevo valor que representa el tamaño funcional y los requerimientos técnicos combinados [21].

Con la traza realizada de las diferencias entre los dos métodos, se plantea en la investigación utilizar el Método Mark II como propuesta para calcular la estimación de esfuerzo en la gestión de proyecto del Expediente de Proyecto basado en Métrica 3.

2.3 Roles y Responsabilidades de Métrica V3

MÉTRICA Versión 3 ha sido concebida para abarcar el desarrollo completo de Sistemas de Información sea cual sea su complejidad y magnitud, por lo cual su estructura y los perfiles de los participantes que intervienen deberán adaptarse y dimensionarse en cada momento de acuerdo a las características particulares de cada proyecto [9].

2.3.1 Roles Principales

- ❖ Perfil Directivo
- ❖ Perfil Jefe de Proyecto
- ❖ Perfil Consultor
- ❖ Perfil Analista
- ❖ Perfil Programador

Para cada uno de estos perfiles se analizan una serie de características importantes a la hora de delimitar su participación en el proyecto [9]:

- ❖ Correspondencia con participantes de MÉTRICA Versión 3.
- ❖ Responsabilidades o funciones a desempeñar en cada uno de los procesos.
- ❖ Perfil o características propias de cada uno de los participantes.

2.3.1.1 Perfil Directivo

El perfil requerido para este grupo de participantes incluye a personas con un nivel alto en la dirección de la organización, conocimiento de los objetivos estratégicos y de negocio que se persiguen y autoridad para validar y aprobar cada uno de los procesos realizados durante el desarrollo del Sistema de Información, además deben tener un conocimiento del entorno y de la organización suficiente para proporcionar, a lo

largo de todo el proyecto, unos requisitos del Sistema adecuados, completos y suficientemente importantes como para considerarse en el catálogo definitivo de requisitos [9].

Dentro de esta categoría se agrupan los siguientes participantes:

- ❖ Comité de Dirección
- ❖ Comité de Seguimiento
- ❖ Directores de usuarios
- ❖ Usuarios expertos

Estos participantes intervienen en todos los procesos de MÉTRICA Versión 3, siendo sus principales responsabilidades y funciones similares, si bien dependiendo del proceso estas pueden experimentar pequeñas variaciones.

Es responsabilidad del Comité de Dirección proveer los recursos necesarios para el cumplimiento de los objetivos propuestos, revisar y aprobar formalmente cada uno de los procesos. Este Comité supone la implicación directa de la alta dirección de la organización en el proyecto, si bien su constitución variará en función de las características del mismo.

Los Directores de las áreas organizativas y de usuarios afectadas por el proyecto aportan información sobre las necesidades planteadas y validan los resultados con el fin de garantizar la identificación, comprensión e incorporación de todos los requisitos con las prioridades adecuadas. Esta misma función la desempeñan con mayor nivel de detalle los usuarios expertos de nivel directivo.

El seguimiento y control del desarrollo del proyecto es responsabilidad del Comité de Seguimiento, que se ocupará de resolver cualquier contingencia que pueda presentarse durante la ejecución del mismo y asegurará la disponibilidad de recursos humanos con los perfiles adecuados y su participación en las actividades donde es necesaria su colaboración [9].

Participantes y actividades que realizan

Participante	Actividad
Comité de Dirección	Plantilla Catálogo de Requisitos
Comité de Seguimiento	Plantilla Registro de Recursos

Tabla 12. Participantes del perfil Directivo y actividades que realizan

2.3.1.2 Perfil Jefe de Proyecto

Este perfil ejerce labores de coordinación y dirección de equipos humanos especializados en la realización de actividades propias de un proceso o interfaz de MÉTRICA Versión 3. La figura principal es el Jefe de Proyecto, el cual recibe el apoyo de los distintos responsables durante la realización de procesos o determinadas actividades a lo largo del proyecto.

El Jefe de Proyecto realiza la estimación del esfuerzo necesario para llevar a cabo el proyecto, selecciona la estrategia de desarrollo, determina la estructura del mismo seleccionando los procesos principales de MÉTRICA Versión 3 que lo integran, fija el calendario de hitos y entregas y establece la planificación del proyecto. Es el encargado de dirigir el proyecto, realizando las labores de seguimiento y control del mismo, revisión y evaluación de resultados y coordinación del equipo de proyecto. Se ocupa también de la gestión y resolución de incidencias que puedan surgir durante el desarrollo del proyecto así como de la actualización de la planificación inicial [9].

Entre sus funciones se encuentran la elaboración de los informes de seguimiento y el archivo de la documentación de gestión del proyecto una vez que este ha finalizado.

Dentro de este perfil se engloban los participantes que se detallan a continuación:

- ❖ Jefe de Proyecto
- ❖ Responsable de Implantación

- ❖ Responsable de Mantenimiento
- ❖ Responsable de Operación
- ❖ Responsable de Sistemas
- ❖ Responsable de Seguridad
- ❖ Responsable de Calidad
- ❖ Responsable de Gestión de Configuración

Los Responsables de Implantación, Operación, Sistemas y Mantenimiento intervienen en procesos principales de MÉTRICA Versión 3, ofreciendo apoyo al Jefe de Proyecto durante la realización de sus actividades. Poseen mayor conocimiento de los aspectos organizativos y de procedimiento habituales en la organización en sus áreas de responsabilidad concretas, facilitando el desarrollo de los procesos que afectan a esas áreas. Aseguran la disponibilidad de los recursos necesarios y la participación activa del equipo humano que coordinan.

Los Responsables de Seguridad y Calidad aportan información relativa a las normas y procedimientos habituales en la organización, completándolos en su caso de acuerdo con los requerimientos particulares del sistema en colaboración con el Jefe de Proyecto. Ofrecen asesoramiento sobre todos los aspectos de seguridad y calidad relativos tanto al producto como al proceso seguido para su obtención, analizando los riesgos y determinando las medidas de control oportunas. Coordinan a los integrantes del Equipo de Seguridad y el Grupo de Aseguramiento de la Calidad [9].

Participantes y Actividades que realizan

Participante	Actividad
Jefe de Proyecto	Plantilla Planificación General del Proyecto
Jefe de Proyecto	Plantilla Documento Visión
Jefe de Proyecto	Plantilla Minuta de Reunión
Jefe de Proyecto	Plantilla Entorno Tecnológico
Jefe de Proyecto	Plantilla Roles y Responsabilidades
Jefe de Proyecto	Plantilla Registro de Recursos
Jefe de Proyecto	Plantilla Estimación de Esfuerzo

Jefe de Proyecto	Plantilla Glosario de Términos
Jefe de Proyecto	Plantilla Plan de Mitigación de Riesgos
Responsable de Seguridad	Plantilla Plan de Seguridad
Responsable de Calidad	Plantilla Plan de Aseguramiento de la Calidad
Responsable de Calidad	Plantilla Plan de Pruebas
Responsable de Calidad	Plantilla Plan de Mediciones
Responsable de Calidad	Plantilla Listas de Chequeo
Responsable de Calidad	Plantilla No conformidades
Responsable de Implantación	Plantilla Plan de Implantación
Responsable de Implantación	Plantilla Plan de Implantación

Tabla 13. Participantes del perfil Jefe de Proyecto y Actividades que realizan

2.3.1.3 Perfil Consultor

La principal función de los Consultores es asesorar en las cuestiones sobre las que tienen un conocimiento especializado. Se diferencia así entre Consultor, que asesora en los aspectos relativos al negocio y Consultor Informático, con un nivel de especialización mayor en los aspectos relacionados con la informática, su aplicación e integración en la organización.

En este perfil se incluyen los siguientes participantes:

- ❖ Consultor
- ❖ Consultor Informático
- ❖ Consultor de las Tecnologías de la Información
- ❖ Consultor de Sistemas de Información
- ❖ Especialista en Comunicaciones
- ❖ Técnico de Sistemas
- ❖ Técnicos de Comunicaciones

En el ámbito de la Consultoría Informática se distingue entre Tecnologías de la Información y Sistemas de Información. El Consultor en Tecnologías de la Información aporta un mayor conocimiento de las últimas

tecnologías, colabora en la evaluación de distintas alternativas tecnológicas y participa en la validación y selección de la solución más adecuada para el sistema a desarrollar, mientras que el Consultor de Sistemas de Información ofrece una opinión experta, pericia o conocimientos relativos a los requisitos del negocio, técnicos y de usuario que han de tenerse en cuenta en el desarrollo de un sistema de información.

Los Técnicos y Especialistas en Sistemas y Comunicaciones cuentan con una visión más precisa de la tecnología existente en la actualidad en la organización o que se valora incorporar, en cuanto a sus requerimientos técnicos, entorno e infraestructura que precisan, implantación, integración con otros sistemas existentes, configuración y pruebas. Aportan su conocimiento y experiencia práctica a la hora de valorar alternativas tecnológicas para el sistema de información, participando activamente durante su implantación y puesta en producción [9].

Participantes y actividades que realizan

Participante	Actividad
Consultor de las Tecnologías de la Información	Plantilla Alternativas de Solución

Tabla 14. Participantes del perfil Consultor y actividades que realizan

2.3.1.4 Perfil Analista

La responsabilidad de los Analistas es elaborar un catálogo detallado de requisitos que permita describir con precisión el sistema de información, para lo cual mantendrán entrevistas y sesiones de trabajo con los responsables de la organización y usuarios, actuando del interlocutor entre estos y el equipo de proyecto en lo que a requerimientos se refiere. Estos requisitos permiten a los analistas elaborar los distintos modelos que sirven de base para el diseño, obteniendo los modelos de datos y de procesos en el caso del

análisis estructurado. Así mismo realizan la especificación de las interfaces entre el sistema y el usuario [9].

En el perfil de Analista se agrupan los siguientes participantes de MÉTRICA Versión 3:

- ❖ Analista
- ❖ Administrador de Bases de Datos
- ❖ Equipo de Arquitectura
- ❖ Equipo de Formación
- ❖ Equipo de Implantación
- ❖ Equipo de Operación
- ❖ Equipo de Seguridad
- ❖ Equipo de Soporte Técnico
- ❖ Equipo de Proyecto
- ❖ Grupo de Aseguramiento de la Calidad

El Administrador de Bases de Datos participa en la obtención del diseño físico de datos, definiendo la estructura física de datos que utilizará el sistema a partir del modelo lógico de datos normalizado o del modelo de clases, teniendo presentes las características específicas del sistema de gestión de base de datos concreto a utilizar, los requisitos establecidos para el sistema de información, y las particularidades del entorno tecnológico, se consiga una mayor eficiencia en el tratamiento de los datos. Si se va a realizar una migración de datos colabora con el equipo de proyecto estimando los volúmenes de las estructuras de datos implicadas, definiendo los mecanismos de migración y carga inicial de datos y participando activamente en su realización. Una vez que el sistema está en producción se ocupa de la gestión y operativa asociada a las bases de datos y al software en el que están implementadas.

Los integrantes del Equipo de Proyecto participan a lo largo de todo el proceso de desarrollo y mantenimiento del sistema de información, si bien su composición puede ir variando en función de las características del proyecto y del proceso que se esté realizando, diferenciando así los Equipos de

Implantación, Operación, Mantenimiento, Arquitectura, Soporte Técnico y Seguridad, coordinados por un Responsable de Equipo, cuyas funciones y perfiles están más especializadas para la realización de un proceso o interfaz concreto [9].

El Equipo de Formación es el encargado de preparar e impartir la formación al equipo responsable de la implantación y operación del sistema, para lo cual se encarga de elaborar un plan de formación que incluye los cursos de formación y sus contenidos, así como los recursos humanos y de infraestructura para llevarlo a cabo. Igualmente define el contenido de la formación que deberá recibir el usuario final del sistema, realizando su seguimiento.

El Grupo de Aseguramiento de la Calidad, dirigido por el Responsable de Calidad, desarrolla el plan de aseguramiento de calidad específico para el proyecto, reflejando en dicho plan entre otros aspectos las actividades de calidad a realizar (normales o extraordinarias). Participa en la revisión de los productos seleccionados para determinar si son conformes o no a los procedimientos, normas o criterios especificados, comprobando que se han llevado a cabo las medidas preventivas o correctoras necesarias. Este grupo es completamente independiente del equipo de proyecto [9].

Participantes y actividades que realizan

Participante	Actividad
Analista	Plantilla Catálogo se Requisitos del PSI
Analista	Plantilla Alternativas de Solución
Analista	Plantilla Plan de Gestión de Requisitos
Analista	Plantilla Plan de Migración
Analista	Plantilla Arquitectura de Información
Administrador de Bases de Datos	Plantilla Plan de Migración
Administrador de Bases de Datos	Plantilla Entorno de Construcción
Equipo de Arquitectura	Plantilla Documento de Arquitectura del Sistema
Equipo de Implantación	Plantilla Plan de Implantación
Equipo de Implantación	Plantilla Plan de Implantación
Equipo de Seguridad	Plantilla Plan de Seguridad

Equipo de Seguridad	Plantilla Documento de Arquitectura del Sistema
Grupo de Aseguramiento de la Calidad	Plantilla Plan de Aseguramiento de la Calidad
Grupo de Aseguramiento de la Calidad	Plantilla Plan de Mediciones
Grupo de Aseguramiento de la Calidad	Plantilla Listas de Chequeo
Grupo de Aseguramiento de la Calidad	Plantilla No Conformidades
Grupo de Aseguramiento de la Calidad	Plantilla Plan de Pruebas
Equipo de Soporte Técnico	Plantilla Documento de Arquitectura del Sistema
Equipo de Proyecto	Plantilla Entorno de Construcción

Tabla 15. Participantes del perfil Analista y actividades que realizan

2.3.1.5 Perfil Programador

Dado que la participación y funciones de los programadores son concretas y limitadas a los procesos de Construcción y Mantenimiento de Sistemas de Información, el perfil de Programador hace referencia únicamente al participante Programador descrito en MÉTRICA Versión 3.

La función del programador, miembro del equipo de proyecto, es construir el código que dará lugar al producto resultante en base al diseño técnico realizado por el analista o analista programador, generando también el código asociado a los procedimientos de migración y carga inicial de datos. Igualmente se encarga de la realización de las pruebas unitarias y participa en las pruebas de conjunto de la aplicación [9].

2.4 Conclusiones del Capítulo

En este capítulo se realizó la propuesta del Expediente de Proyecto basado en la metodología Métrica V3, la cual quedó conformada para tres tipos de proyectos, debido a los tipos de línea que abarca el Área Temática Herramientas para la Teleformación, Desarrollo, Servicios de Personalización e Instalación. Para la formulación de este expediente se tuvo en cuenta aspectos como: procesos e interfaces de Métrica V3,

requisitos y estructura de la misma. Se elaboró además, una descripción detallada de los participantes que propone Métrica V3 y las actividades que estos realizan, para mantener la organización y planificación de los sistemas de información. Posteriormente se analizaron los métodos que brinda la metodología seleccionada para el cálculo de Puntos de Función en el desarrollo estructurado, Albrecht y Mark II, destacando de cada uno de ellos sus características, surgimiento, procedimientos para el cálculo de Puntos de Función, y por último las diferencias y similitudes que ambos presentan, con el objetivo de optimizar la propuesta final del Método para el cálculo de Puntos de Función a utilizar en la plantilla Estimación de Esfuerzo perteneciente a la interfaz Gestión de Proyecto.

CAPITULO 3: APLICACIÓN Y ANÁLISIS DE RESULTADOS

Introducción

En el presente capítulo se aplica el Expediente de Proyecto de Instalación propuesto en el Capítulo 2 a un proyecto productivo del Área Temática Herramientas para la Teleformación, realizándose un análisis de los resultados alcanzados, utilizando para dicho análisis métodos como encuestas realizadas a miembros del equipo de implantación perteneciente al Área Temática que utilicen programación estructurada.

3. Aplicación del Expediente de Proyecto de Instalación

Con el objetivo de aplicar la propuesta de Expediente de Proyecto para proyectos de instalación se ejecutó como caso de prueba la instalación de la plataforma Moodle, *Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular; paquete de software para la creación de cursos y sitios web, que permite el diseño de cursos estructurados incluyendo recursos y actividades, y llevar un seguimiento detallado de las acciones del educando y sus avances, cuestiones necesarias para dirigir un buen proceso de aprendizaje en línea* [22]. La instalación de esta plataforma se realizó para la Dirección de Teleformación de la UCI.

Para lograr una satisfactoria instalación de la plataforma Moodle, el equipo de trabajo lleva a cabo un Plan de Implantación, el cual a su vez se divide en tres fases.

3.1 Fase de Definición

La fase de definición recoge todas las peticiones del usuario, la forma en que se gestionó la prestación del servicio, el Documento Visión y la Solución propuesta. Se establece la comunicación directa con el cliente, solicitando los requisitos previos para la instalación de la aplicación. Además, se analizaron puntos como

las características de la aplicación a instalar, los involucrados en la instalación de la misma, el sistema de soporte al cliente y los beneficios que trae la instalación de la aplicación para el mismo.

Las plantillas que recogen todo el contenido que abarca la primera fase son:

- ❖ Solicitud
- ❖ Requisitos Necesarios para la Instalación
- ❖ Documento Visión
- ❖ Lista de Riesgo
- ❖ Minutas de Reunión
- ❖ Roles y Responsabilidades

Otro de los aspectos fundamentales en la primera fase son las características de la infraestructura para soportar la aplicación que se necesita instalar, destacando entre otros puntos si el sistema se divide en capas, si hay recursos para soportar la cantidad de usuarios, la cantidad de servidores disponibles, si se dispone de un servidor no dedicado (que esté siendo utilizado por otras aplicaciones).

Como segundo paso de la fase inicial de la instalación se realizó el Documento Visión el cual tiene como objetivo plasmar el alcance o la visión del proyecto. En él se trataron aspectos relacionados con los usuarios involucrados con la plataforma Moodle, las oportunidades de negocio que serían los posibles mercados, dentro de ellos se encuentran todas las universidades, centros educacionales, empresas e instituciones que realicen actividades de capacitación contando con el LMS Moodle.

Después de realizado el Documento Visión, el responsable del equipo de implantación y el jefe de proyecto muestran al cliente la solución propuesta para instalar, dicha propuesta es valorada por el cliente, y si la propuesta es aceptada satisfactoriamente, el responsable y el equipo de implantación, pasan a la instalación de la misma.

En esta fase se tuvo presente los riesgos que pudieran afectar la correcta instalación de la plataforma Moodle según el tiempo estimado en el cronograma de trabajo. Para ello desarrolló una lista de riesgos la cual describe cómo identificar, estudiar y eliminar las fuentes de riesgos que puedan amenazar la finalización en tiempo y forma de la instalación de la aplicación, así como la probabilidad que tiene cada riesgo de suceder, y la posible solución si llegara a producirse.

3.2 Fase de Instalación

En la segunda fase se recoge información sobre la configuración de servidores, la optimización de la aplicación a instalar para así garantizar un correcto funcionamiento de la misma y una satisfactoria aceptación del cliente.

Las plantillas que recogen todo el contenido que abarca la segunda fase son:

- ❖ Cronograma de Trabajo
- ❖ Minuta de Reunión
- ❖ Diagrama de Despliegue

Otros de los argumentos que comprende esta fase son, el cronograma de instalación, el cual controla la distribución de las actividades, así como el tiempo de duración del proceso de instalación y los involucrados en las mismas. Además, se planifican reuniones con el equipo de implantación, dirigidas a brindar soluciones a posibles errores o dificultades de la instalación. Conjuntamente se procedió con el despliegue y adaptación de la instalación según la solución propuesta, realizando el diagrama de despliegue basándose en la estructura de la aplicación a instalar, dicho diagrama modela la arquitectura en tiempo de ejecución de un sistema, mostrando la configuración de los elementos de hardware (nodos) así como los elementos y artefactos del software trazados en esos nodos.

3.3 Fase de Pruebas

La tercera y última fase de la instalación es el diseño y realización de los distintos tipos de pruebas con el fin de detectar los posibles errores.

Las plantillas que recoge todo el contenido que abarca la tercera fase es:

- ❖ Plan de Pruebas
- ❖ Minutas de Reunión

Las pruebas de funcionalidad encargadas del funcionamiento de la aplicación y la verificación del manejo de roles. Además están dirigidas a asegurar que el sistema de información realice correctamente todas las funciones que se han detallado en las especificaciones dadas por el usuario del sistema.

Las pruebas de estrés son responsables de cuantificar el soporte de la aplicación de acuerdo a la infraestructura, y de optimizar la misma a medida que se realice la prueba.

En las pruebas de seguridad se cerciora que siempre se use un protocolo seguro para la autenticación, además en la misma se procura que los ficheros no estén en un lugar accesible en la web, es decir que cumpla con la estructura Base Dato-Aplicación Web-Ficheros.

Dichas pruebas se encuentran reflejadas en la plantilla Plan de Pruebas, el cual tiene como objetivo definir el proceso formal bajo el cual serán efectuadas las pruebas de la aplicación, a fin de garantizar, con la aceptación formal del cliente, que la aplicación que está siendo implantada no sólo está libre de fallas de funcionalidad técnica, sino que a su vez la misma cubre totalmente los requerimientos exigidos por el cliente, los cuales fueron definidos con los mismos.

Además de contener una descripción de los tipos de pruebas, este plan también aborda aspectos relacionados con los niveles de pruebas, requisitos de entorno de pruebas, perfiles implicados en la

realización de las mismas, cronograma de pruebas, así como el análisis y evaluación de los resultados obtenidos con el desarrollo de las mismas.

3.4 Grado de aceptación del Expediente de Proyecto de Instalación

Para medir la factibilidad del Expediente de Proyecto de instalación se realizó una encuesta (Ver Anexo 2) a los integrantes del equipo de instalación, con el objetivo de detectar los posibles errores de la propuesta de expediente.

Con la aplicación de dicha encuesta se determinaron varios porcentos de aceptación plasmados en gráficos de barras. A continuación se muestran dichos gráficos:

En el siguiente gráfico se representa el porcentaje de aceptación del Expediente de Proyecto de instalación, es decir si dicho Expediente recoge toda la información necesaria para la correcta implantación de un software.

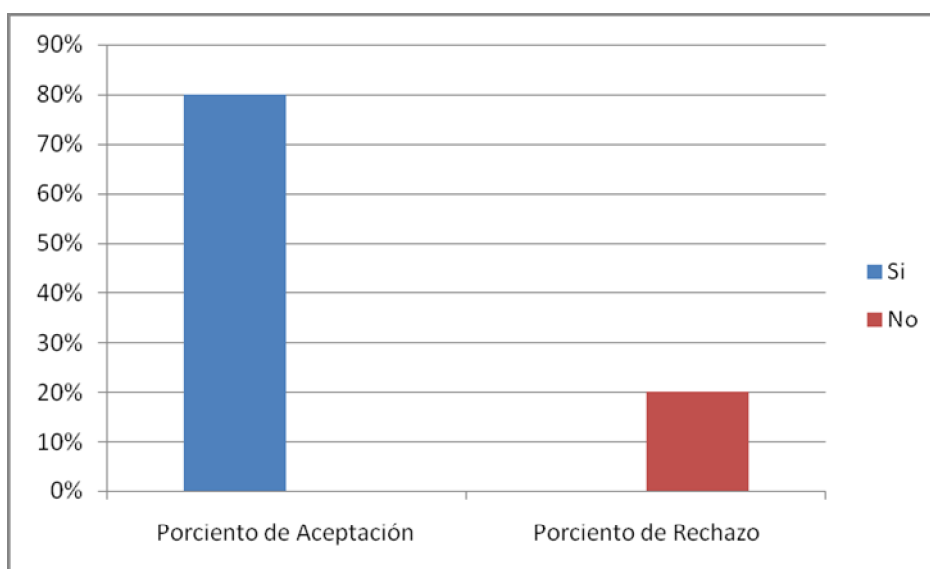


Figura 7: Aceptación del Expediente de Proyecto

El gráfico que se muestra a continuación representa el orden lógico del Plan de Implantación, es decir si en realidad este plan constituye una guía de ejecución para llevar a cabo la instalación.

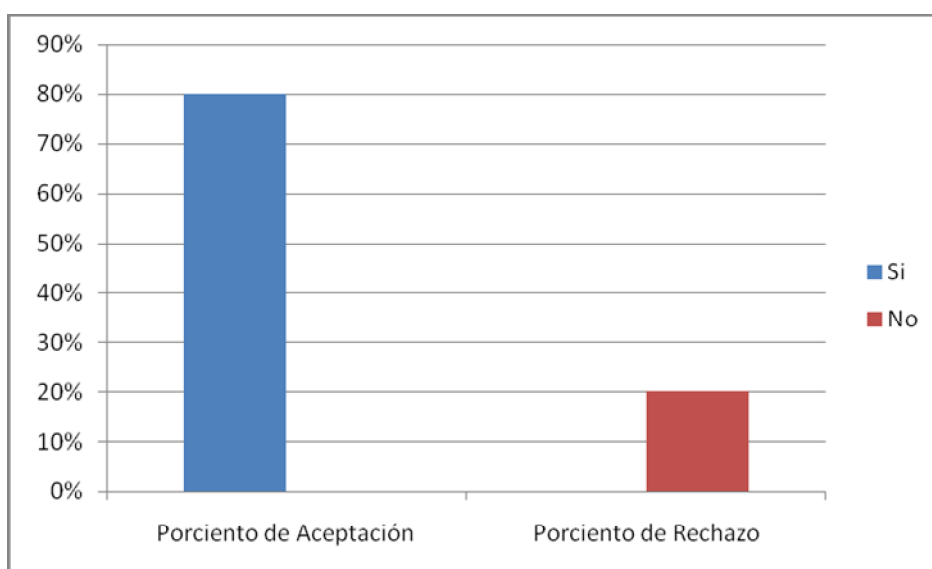


Figura 8: Orden Lógico del Plan de Implantación

En el siguiente gráfico se muestra el porcentaje de factibilidad del Expediente de Proyecto de instalación propuesto.

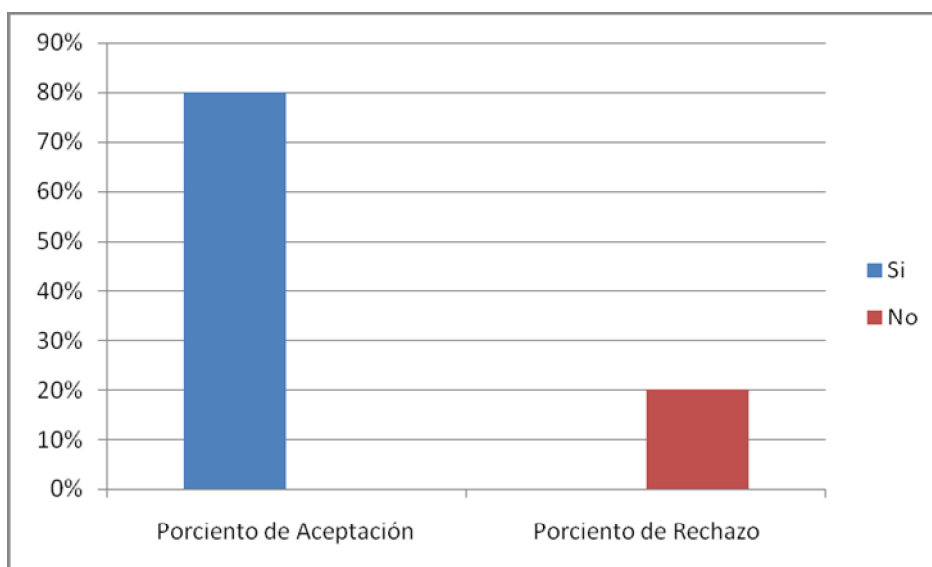


Figura 9: Factibilidad del Expediente de Proyecto

El gráfico que se muestra a continuación visualiza el porcentaje de aceptación de las plantillas que componen el Expediente de Proyecto de Instalación, es decir cuál o cuáles plantillas constituyen mayor importancia para la documentación de los proyectos de instalación.

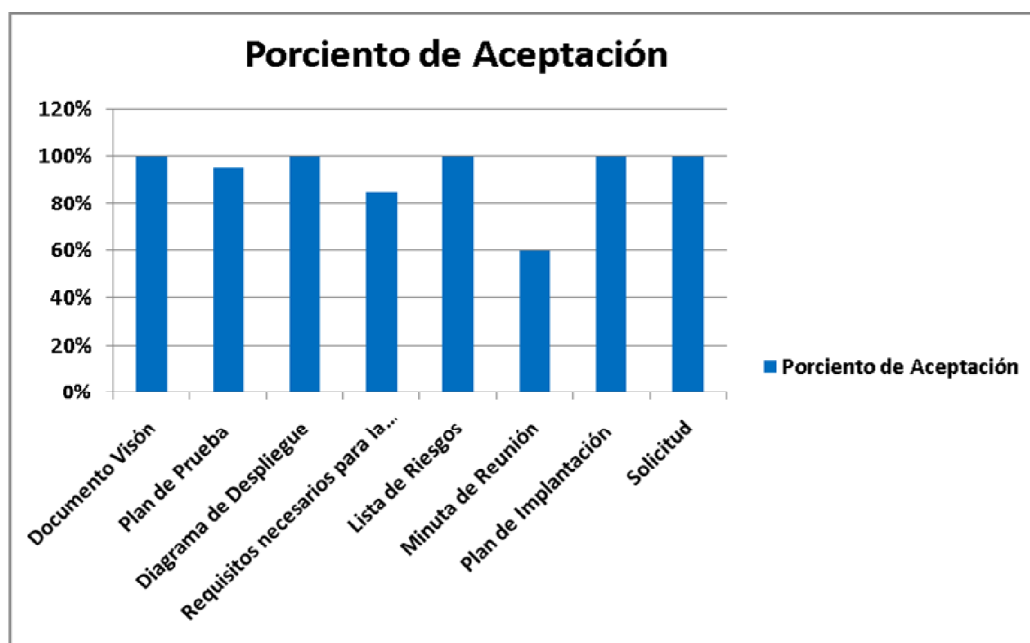


Figura 10: Aceptación de las plantillas

El siguiente Grafico muestra los criterios de los integrantes del equipo de personalización acerca del Expediente de Proyecto de instalación.

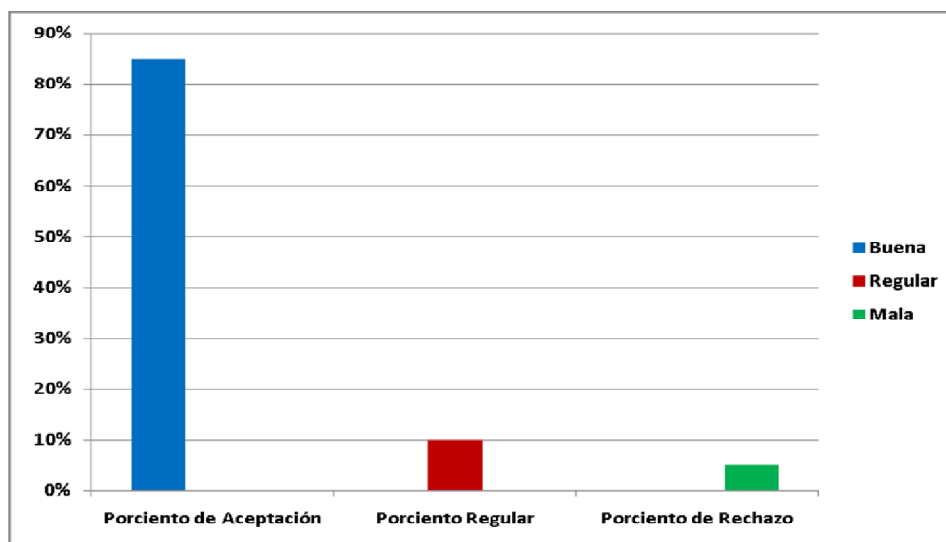


Figura 11: Opinión del Expediente de Proyecto de Instalación

3.5 Conclusiones

Con el desarrollo de este capítulo se demostró que la gestión de instalación realizada cumple todos los requisitos propuestos en la guía de ejecución de instalación, es decir, el Plan de Implantación. Además, el Expediente de Proyecto de Instalación basado en Métrica V3 se ajusta adecuadamente a las necesidades del proyecto. Conjuntamente se arriba a la conclusión que el resultado de la aplicación del Expediente de Proyecto de Instalación fue satisfactorio, con aceptación plena y parcial de los miembros del proyecto encuestados.

CONCLUSIONES GENERALES

A partir de los objetivos planteados en esta investigación donde se estudiaron las principales metodologías estructuradas, para los proyectos productivos pertenecientes al Área Temática Herramientas para la Teleformación de la Facultad 8, se arribó a las siguientes conclusiones:

- ❖ La propuesta de Expediente de Proyecto definido permite agrupar y organizar todos los artefactos que se generan durante el proceso de desarrollo de software de los proyectos que utilizan programación estructurada del Área Temática Herramientas para la Teleformación.
- ❖ El Plan de Implantación propuesto en el Expediente de Proyecto de Instalación constituye una guía de ejecución y permite gestionar el proceso de instalación de un software.
- ❖ La documentación realizada en este trabajo constituye una guía para la gestión de proyectos que utilizan programación estructurada del Área Temática Herramientas para la Teleformación.

RECOMENDACIONES

Como resultado de la investigación y elementos a tener en cuenta, se hacen las siguientes recomendaciones, estas pueden ser desarrolladas en trabajos de diplomas en el curso 2009-2010:

- ❖ Aplicar el Expediente de Proyecto a proyectos de desarrollo y servicios de personalización.
- ❖ Se recomienda migrar de la estructura organizativa de la propuesta de Expediente de Proyecto basado en Métrica V3 realizado en esta investigación, a la estructura organizativa del Expediente de Proyecto definido por la UCI.
- ❖ Impartir un taller de Métrica V3 a profesores y estudiantes antes aplicar la propuesta de Expediente de Proyecto en los proyectos que utilicen programación estructurada.

REFERENCIAS BIBLIOGRÁFICAS

- [1]. **Constantine y Demarco.** *Surgimiento de las metodologías estructuradas, primeros pasos.* 1979.
- [2]. **Constantine, Yourdon &.** *Características de la programación, y diseño estructurado, definición de programación estructurada.* 1977.
- [3]. **Yourdon, Myers &.** *Las especificaciones funcionales y particionadas.* 1975.
- [4]. **Böhm, C, Jacopini, G y Comm.** Teorema Fundamental de la programación estructurada. [En línea] 1966. [Citado el: 10 de enero de 2009.]
<http://sj.fundacionloyola.org/sj/F70/descargas/Uno/ld/K697/TEMA+1-+PROG.+ESTRUCTURADA+Y+ORDINOGRAMAS.pdf>.
- [5]. Surgimiento e historia de Métrica 3. [En línea] [Citado el: 3 de diciembre de 2008.]
http://www.cc.uah.es/hilera/docs/1999/c_jisbd/c_jisbd.htm.
- [6]. **Gane y Sarsons.** *Principales problemas que presentaba el enfoque estructurado a fines de los años 70.* 1977.
- [7]. Descripción general de la metodología Eurométodo. [En línea] [Citado el: 4 de diciembre de 2008.]
<http://www.csi.map.es/csi/pg5e40.htm>.
- [8]. **Auladirectiva.** *Auladirectiva.* 2006.
- [9]. Ministerio de Administraciones Públicas de España. [En línea] 2001. [Citado el: 14 de abril de 2009.]
<http://www.csi.map.es/csi/metrica3/Participantes.pdf>.
- [10]. **Dirección Calidad UCI.** Dirección de Calidad de Software. [En línea] 2008. [Citado el: 5 de marzo de 2009.] <http://calidadsoft.prod.uci.cu/>.
- [11]. **Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico.* La Habana : Felix Varela, 2005.
- [12]. **CMMI.** *The Capability Maturity Model Integrated.* . USA : Carnegie Mellon Software Engineering Institute, 2002.
- [13]. **IEEE.** *Standard for Software Configuration Management "*. s.l. : IEEE Computer Society, 1990.

- [14]. —. " *Standard for Software Quality Assurance Plans*". . s.l. : IEEE Computer Society, 1998.
- [15]. **Humphrey, Watts**. *Introducción al Proceso Software Personal*. s.l. : Pearson Education, S.A, 2001.
- [16]. **CMM**. *The Capability Maturity Model*. USA : Carnegie Mellon Software Engineering Institute, 1993.
- [17]. **RUP**. *Rational Unified Process, IBM*. 2003.
- [18]. Fases de métrica 3 y diferentes versiones. . [En línea] [Citado el: 2 de febrero de 2009.]
<http://www.unap.cl/~setcheve/Metrica/m/index.html>.
- [19]. **Albrecht, A**. " *Measuring Application Development Productivity*". Monterey, CA : IBM Application Development Symposium, 1979.
- [20]. **Symons, C, Wiley, John y Sons**. *Software sizing and estimating Mark II FPA*. 1991.
- [21]. **Symons, C**. *MK II Function Points Analysis Counting Practices Manual, version 1.3.1, UKSMA (United Kingdom Software Metrics Association)*,. 1998.
- [22]. **de Pablos, Gema**. Educación Observatorio Tecnológico. [En línea] Educación Observatorio Tecnológico, 18 de noviembre de 2005. [Citado el: 2009 de mayo de 2009.]
<http://observatorio.cnice.mec.es/>.
- [23]. Unified Modeling Language Documentation. UML Resource Center . [En línea] UML Resource Center , 1999. [Citado el: 15 de marzo de 2009.] • Semejanzas de Métrica 3 con RUP. Disponibles en: Unified Modeling Language <http://www.rational.com/uml/resources/documentation>.
- [24]. Aspectos sobre datos referenciados en la introducción. [En línea] [Citado el: 13 de diciembre de 2008.] http://www.rhernando.net/modules/tutorials/doc/ing/met_soft.html.
- [25]. Clasificación de las metodologías estructuradas. [En línea]
http://static.scribd.com/docs/dsmodb48dz9x5.swf?INITIAL_VIEW=width.
- [26]. Definición de desarrollo estructurado. . [En línea] [Citado el: 8 de diciembre de 2008.]
http://sistemas.itlp.edu.mx/tutoriales/pascal/u1_1_5.html.
- [27]. Definición y características de la programación estructurada, Teorema de la estructura. . [En línea] [Citado el: 14 de enero de 2009.] <http://html.rincondelvago.com/programacion-estructurada.html>.

- [28]. Descripción de las fases de Métrica 3, así como sus versiones anteriores. . [En línea] [Citado el: 1 de diciembre de 2008.] <http://www.lpsi.eui.upm.es/MDes/TfcMetrica/previst.htm>.
- [29]. Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1. [En línea] Ministerio para las Administraciones Públicas. Madrid , 1995. [Citado el: 23 de noviembre de 2008.] <http://www.map.es/csi/pg5m40.htm>..
- [30]. Proyecto MÉTRICA Versión 3. [En línea] Ministerio para las Administraciones Públicas. [Citado el: 23 de noviembre de 2008.] <http://www.map.es/csi/pg5m42.htm>..
- [31]. Surgimiento y características. [En línea] [Citado el: 12 de enero de 2009.] <http://www.monografias.com/trabajos/progestructu/progestructu.shtml>.
- [32]. **Antonio, Angélica de.** *Gestión de Configuración*. Chile : s.n., 2001 .
- [33]. **Booch, G y Rumbaugh, J.** *El Lenguaje Unificado de Modelado*. Madrid : Addison Wesley, 1999.
- [34]. **COCKBUN, A y WILLIAMS, L.** *The Costs and Benefits of Pair Programming*. s.l. : Humans and Technology Technical Report, 2000.
- [35]. **Elisa, Gallo M.** European Software Institute. [En línea] 2006. [Citado el: 5 de diciembre de 2008.] <http://www.esi.es/Berrikuntza>.
- [36]. **FOWLER, M, BECK, K y BRANT, J.** *Refactoring: Improving the Design of Existing Code*. s.l. : Addison-Wesley, 1999.
- [37]. **Grant Rule, P.** "The Importance of the Size of Software Requirements". India : Software Measurement Services, NASSCOM, 2001.
- [38]. **Hilera, J R.** *Metodología MÉTRICA Orientada a Objetos*. . s.l. : NOVÁTICA.
- [39]. **Joyanes Aguilar, L.** "Fundamentos de programación. Algoritmos, estructuras de datos y objetos (3ª edición)". s.l. : McGraw-Hill, 2003.
- [40]. **Lugo Rivero, Mario Alberto.** Sistema de nomina basado en la metodología de Yourdon. [En línea] [Citado el: 3 de diciembre de 2008.] <http://www.fce.unju.edu.ar>.
- [41]. **MARTIN, R.** Continuous Care vs. [En línea] 2002. www.objectmentor.com/resources/articles/Continuous_Care.pdf.

Referencias Bibliográficas

[42]. **Myers, y otros.** *Definición del Módulo del programa, la normalización de la estructura los módulos de programa.* 1980.

[43]. **Myers y Yourdon.** *Las especificaciones funcionales y particionadas.* 1975.

[44]. **PETROSKI, H.** *La Evolución de las Cosas Útiles.* Vintage Brooks. *Comparaciones entre diferentes metodologías.* 1994.

[45]. **PRESSMAN, R.** *Ingeniería de Software un enfoque practico.* 5ta p.

BIBLIOGRAFÍA

Albrecht, A. *"Measuring Application Development Productivity"*. Monterey, CA : IBM Application Development Symposium, 1979.

Andrés, Maria Angeles Sanchez. 2002. *Programación Estructurada y Fundamentos de Programación*. España : s.n., 2002. 844815575.

Antonio, Angélica de. *Gestión de Configuración*. Chile : s.n., 2001 .

Areba, J. Barranco de. 2001. *Metodología del Análisis Estructurado de Sistemas*. 2001.

Auladirectiva. *Auladirectiva*. 2006.

Aspectos sobre datos referenciados en la introducción. [En línea] [Citado el: 13 de diciembre de 2008.] http://www.rhernando.net/modules/tutorials/doc/ing/met_soft.html.

BALAKISNAN, S. K., G. *Resolving Gathering of User Requirements – A Lightweight Methodology Approach in the Development of Medium Maturity Level Web Sites*, 2006. [2006]. Disponible en: <http://www.ucti.edu.my/wps/issue1/wp-06-04-paper.pdf>

Böhm, C, Jacopini, G y Comm. Teorema Fundamental de la programación estructurada. [En línea] 1966. [Citado el: 10 de enero de 2009.] <http://sj.fundacionloyola.org/sj/F70/descargas/Uno/Id/K697/TEMA+1-+PROG.+ESTRUCTURADA+Y+ORDINOGRAMAS.pdf>.

Booch, G y Rumbaugh, J. *El Lenguaje Unificado de Modelado*. Madrid : Addison Wesley, 1999.

Clasificación de las metodologías estructuradas. [En línea] http://static.scribd.com/docs/dsmodb48dz9x5.swf?INITIAL_VIEW=width.

CMMI. *The Capability Maturity Model Integrated*. . USA : Carnegie Mellon Software Engineering Institute, 2002.

CMM. *The Capability Maturity Model*. USA : Carnegie Mellon Software Engineering Institute, 1993.

COCKBUN, A y WILLIAMS, L. *The Costs and Benefits of Pair Programming*. s.l. : Humans and Technology Technical Report, 2000.

Constantine y Demarco. *Surgimiento de las metodologías estructuradas, primeros pasos*. 1979.

Constantine, Yourdon &. *Características de la programación, y diseño estructurado, definición de programación estructurada.* 1977.

De Pablos, Gema. Educación Observatorio Tecnológico. [En línea] Educación Observatorio Tecnológico, 18 de noviembre de 2005. [Citado el: 2009 de mayo de 2009.] <http://observatorio.cnice.mec.es/>.

Definición de desarrollo estructurado. . [En línea] [Citado el: 8 de diciembre de 2008.] http://sistemas.itlp.edu.mx/tutoriales/pascal/u1_1_5.html.

Definición y características de la programación estructurada, Teorema de la estructura. . [En línea] [Citado el: 14 de enero de 2009.] <http://html.rincondelvago.com/programacion-estructurada.html>.

Descripción de las fases de Métrica 3, así como sus versiones anteriores. . [En línea] [Citado el: 1 de diciembre de 2008.] <http://www.lpsi.eui.upm.es/MDes/TfcMetrica/previst.htm>.

Descripción general de la metodología Eurométodo. [En línea] [Citado el: 4 de diciembre de 2008.] <http://www.csi.map.es/csi/pg5e40.htm>.

Dirección Calidad UCI. Dirección de Calidad de Software. [En línea] 2008. [Citado el: 5 de marzo de 2009.] <http://calidadsoft.prod.uci.cu/>.

Elisa, Gallo M. European Software Institute. [En línea] 2006. [Citado el: 5 de diciembre de 2008.] <http://www.esi.es/Berrikuntza>.

Fases de métrica 3 y diferentes versiones. . [En línea] [Citado el: 2 de febrero de 2009.] <http://www.unap.cl/~setcheve/Metrica/m/index.html>.

Fernando José Pérez, Francisco Javier Marín . 2008. Xuletas: Metodologías Estructuradas. *Xuletas: Metodologías Estructuradas.* [En línea] 2008. [Citado el: 10 de 12 de 2009.] <http://www.xuletas.es/ficha/metodologias-estructuradas>.

FOWLER, M, BECK, K y BRANT, J. *Refactoring: Improving the Design of Existing Code.* s.l. : Addison-Wesley, 1999.

Gane y Sarsons. *Principales problemas que presentaba el enfoque estructurado a fines de los años 70.* 1977.

Grant Rule, P. "The Importance of the Size of Software Requirements". India : Software Measurement Services, NASSCOM, 2001.

Hans Mahuke. *Ingeniería del Software Práctico y Conciso.* s.l. : Prentice Hall.

Hernando Roberto. 2004. rhernando_Tutoriales. *rhernando_Tutoriales*. [En línea] 09 de Marzo de 2004. <http://www.rhernando.net/modules/tutorials/>.

Hilera, J R. *Metodología MÉTRICA Orientada a Objetos*. . s.l. : NOVÁTICA.

Humphrey, Watts. *Introducción al Proceso Software Personal*. s.l. : Pearson Education, S.A, 2001.

IEEE. *Standard for Software Configuration Management* ". s.l. : IEEE Computer Society, 1990.

J. M Juran. 1985. *Juran y la Calidad por el Diseño*. s.l. : Diaz de Santo, 1985.

Jean Patrick Matheron. *MERISE: Metodología de desarrollo de sistemas (teoría aplicada)*. s.l. : Paraninfo.

Joyanes Aguilar, L. *"Fundamentos de programación. Algoritmos, estructuras de datos y objetos (3ª edición)"*. s.l. : McGraw-Hill, 2003.

Kenneth C. Laudon, Jane P. Laudon. 2005. *Sistemas de Información General*. s.l. : Prentice Hall, 2005.

Lugo Rivero, Mario Alberto. Sistema de nomina basado en la metodología de Yourdon. [En línea] [Citado el: 3 de diciembre de 2008.] <http://www.fce.unju.edu.ar>.

Maria Dolores Alansa Sanlas, Silvia Rumew. 1994. *Metodología De La Programación. Programación Estructurada*. s.l. : Pretice Hall, 1994.

Mario G. Piattini. Analisis y Diseño detallado de aplicaciones informáticas y de Gestión. [aut. libro] Mario G. Piattini. *Analisis y Diseño detallado de aplicaciones informáticas y de Gestión*. s.l. : RAMA.

MARTIN, R. Continuos Care vs. [En línea] 2002. www.objectmentor.com/resources/articles/Continuous_Care.pdf.

Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1. [En línea] Ministerio para las Administraciones Públicas. Madrid , 1995. [Citado el: 23 de noviembre de 2008.] <http://www.map.es/csi/pg5m40.htm>..

Ministerio de Administraciones Públicas de España. [En línea] 2001. [Citado el: 14 de abril de 2009.] <http://www.csi.map.es/csi/metrica3/Participantes.pdf>.

Myers, y otros. *Definición del Módulo del programa, la normalización de la estructura los módulos de programa*. 1980.

Myers y Yourdon. *Las especificaciones funcionales y particionadas*. 1975.

PETROSKI, H. *La Evolución de las Cosas Útiles.* Vintage Brooks. Comparaciones entre diferentes metodologías. 1994.

Piattini, Mario G. Analisis y Diseño detallado de Gestión. [aut. libro] Guillermo Montoya Greogorio Cabrera. *Analisis y Diseño detallado de Aplicaciones informáticas y de Gestión.* s.l. : Mc Graw Hill.

Piña, Ramon Antonio Rodriguez. 2006. Acimed. *Metodología para el análisis de información orientada al análisis de tendencias en el Web superficial a partir de fuentes no estructuradas.* [En línea] 2006. http://bvs.sld.cu/revistas/aci/vol14_6_06/aci06606.htm.

PRESSMAN, R. *Ingeniería de Software un enfoque practico.* 5ta p.

Pressman Rogers. *Ingenieria del Software: Un enfoque Práctico.* s.l. : Mc Graw Hill.

Pressman, Roger. *Ingeniería del Software. Un enfoque práctico.* La Habana : Felix Varela, 2005.

Proyecto MÉTRICA Versión 3. [En línea] Ministerio para las Administraciones Públicas. [Citado el: 23 de noviembre de 2008.] <http://www.map.es/csi/pg5m42.htm>..

Quero Catalinas Enrique. 2003. *Programación En Lenguajes Estructurados.* s.l. : Paraninfo, 2003. 8497320034.

RUP. *Rational Unified Process, IBM.* 2003.

Rumbaugh, Jacobson Booch. 1999. *El Proceso Unificado de desarrollo de Software.* s.l. : Addison Wesley, 1999.

Seco Amescua y otros. 2003. *Analisis y Diseño Estructurado y Orientado a Objeto de Sistemas Informáticos.* España : s.n., 2003. 9788448139247.

" *Standard for Software Quality Assurance Plans*". . s.l. : IEEE Computer Society, 1998.

Surgimiento e historia de Métrica 3. [En línea] [Citado el: 3 de diciembre de 2008.] http://www.cc.uah.es/hilera/docs/1999/c_jisbd/c_jisbd.htm.

Surgimiento y características. [En línea] [Citado el: 12 de enero de 2009.] <http://www.monografias.com/trabajos/progestructu/progestructu.shtml>.

Symons, C, Wiley, John y Sons. *Software sizing and estimating Mark II FPA.* 1991.

Symons, C. *MK II Function Points Analysis Counting Practices Manual, version 1.3.1, UKSMA (United Kingdom Software Metrics Association),.* 1998.

Unified Modeling Language Documentation. UML Resource Center . [En línea] UML Resource Center , 1999. [Citado el: 15 de marzo de 2009.] • Semejanzas de Métrica 3 con RUP. Disponibles en: Unified Modeling Languag <http://www.rational.com/uml/resources/documentation>.

Yourdon Edward. *Análisis EStructurado Moderno*. s.l. : Prentice Hall.

Yuordon Edward. *Análisis Estructurado Moderno*. [aut. libro] Edward Yuordon. *Análisis Estructurado Moderno*. s.l. : Prentice Hall.

Yourdon, Myers &. *Las especificaciones funcionales y particionadas*. 1975.

ANEXOS

1 Cronograma de Trabajo Soporte de Herramientas para la Teleformación



Cronograma de Trabajo Soporte de herramientas para la Teleformación

Este cronograma de trabajo está desarrollado en presencia y conformidad de las partes y debe cumplirse si no hay factores externos que intervengan.



Acción	F. Inicio	F. Finalización	Descripción	Responsable(s)
Documento Visión	<18/05/09>	<20/05/09>	Se describe el alcance o la visión del proyecto	Yordanis Gutiérrez
Lista de Riesgos	<19/05/09>	<20/05/09>	Se seleccionan los posibles riesgos que pueden ocurrir en el transcurso de la instalación.	Reynier Palacio
Minuta de Reunión	<22/05/09>	<22/05/09>	Reunión con los integrantes del equipo de instalación.	Yordanis Gutiérrez
Diagrama de Despliegue	<25/05/09>	<25/05/09>	Descripción de nodos	Reynier Palacio

2 Plan de Pruebas

2. Objetivos de las Pruebas del Sistema

El presente documento de definición de PLAN DE PRUEBAS, tiene como objetivo definir el proceso formal bajo el cual serán efectuadas las pruebas de la aplicación, a fin de garantizar, con la aceptación formal del cliente, que la aplicación que está siendo implantada no sólo está libre de fallas de funcionalidad técnica sino que a su vez, la misma cubre totalmente los requerimientos exigidos por el cliente, los cuales fueron definidos con los clientes.

Bajo estas premisas, el presente documento presentará el resumen de las pruebas a realizar para garantizar la cobertura total de cada uno de los requerimientos y las fechas en las cuales cada una de estas pruebas serán ejecutadas tanto con los integrantes del equipo de desarrollo y los analistas de pruebas, quién deberá validarlas y aceptarlas como exitosa a través de un formato de aceptación de pruebas definido para el proyecto.

Definición de los Requisitos del Entorno de Prueba

- **Requisitos básicos de hardware y software base: sistemas operativos, gestores de bases de datos, monitores de teleproceso, etc.**

Para la ejecución del plan de pruebas de la aplicación se contarán con los siguientes recursos disponibles:

RECURSO DE SOFTWARE:

Para la ejecución de las pruebas se utilizará el Mozilla e Internet Explorer

RECURSO DE HARDWARE:

Para la ejecución del plan de las pruebas se utilizara el servidor que es el actual servidor de desarrollo

SISTEMA OPERATIVO

El servidor de desarrollo esta montado sobre GNU/Linux Debian 4.0

GESTOR DE BASE DE DATOS

La prueba se realizara sobre un servidor de base de datos montado con PostgreSQL 8.1

- **Requisitos de configuración de entorno: librerías, bases de datos, ficheros, procesos, comunicaciones, necesidades de almacenamiento, configuración de accesos, etc.**

Los requisitos de configuración de entorno que se deben tomar en cuenta para el buen funcionamiento de la aplicación son los siguientes:

- El servidor debe tener instalado PHP5 con las siguientes librerías:
 - [Mysql](#)
 - [Dev](#)
 - [Curl](#)
 - [Gd](#)
 - [xmlrpc](#)
 - [mbstring](#)
 - [openssl](#)
 - [tokenizer](#)
 - [ctype](#)
 - [iconv](#)

Toda la información de la aplicación se guardara en el servidor donde se encuentra montada también la base de datos de la aplicación.

4. Especificación de niveles de prueba

4.1 Productos a entregar como resultado de la ejecución de las pruebas.

⊕ Plataforma Moodle para la dirección de Teleformación es la aplicación que se ejecutará con su correcto funcionamiento después que le hayan aplicado todas las pruebas, ejemplo Unitarias, integración, sistemas, implantación y aceptación.

□

Pruebas a realizar por requerimientos

A continuación se presenta el plan de pruebas a ejecutar para garantizar la total cobertura de los requerimientos. Para cada una de las pruebas mencionadas, se definirán los diferentes casos de pruebas necesarios e indispensables para completar todas las pruebas requeridas por la aplicación y de ésta forma garantizar la cobertura total de los requerimientos de acuerdo a las especificaciones de calidad exigidas por el cliente.

Pruebas de Página de Autenticación (Intranet)

En este módulo los aspectos a probar son:

- Verificación de Usuario y Clave con el del directorio activo.
- Verificación de Rol y tipo de Usuario.
- Verificar que aparezca el nombre del usuario una vez autenticado
- Bloqueo de Usuario 3 intentos fallidos

Pruebas del Menú Principal:

En este módulo los aspectos a probar son:

- Verificación de las opciones del menú de acuerdo con el perfil del usuario.
- Verificar el funcionamiento del link de cerrar sesión.
- Verificar el funcionamiento del link de Edupedia
- Verificar el funcionamiento del link de Catálogo de Programas de Aprendizaje
- Verificar el funcionamiento del link de Cursos

Pruebas del Catálogo de Programas de Aprendizaje:

En este módulo los aspectos a probar son:

- Probar el acceso a este Catálogo a través del link en la página principal.
- Verificar que funcione el botón "Buscar Curso"
- Verificar que a parezca una lista larga con todos los cursos disponibles
- Verificar que al seleccionar cualquier curso se despliegue con todos sus componentes (texto, imágenes, etc.)

Pruebas del módulo Cursos

En este módulo los aspectos a probar son:

- Funcionamiento del link al Módulo Categorías
- Despliegue del Menú "Lista General"
- Despliegue del Menú "Mis cursos"

4.3 Planificación Temporal

A continuación se presenta el detalle de las fechas de ejecución de las pruebas y el módulo o funcionalidad a probar durante la fecha especificada.



Módulo a Probar	Fecha Prueba Unitaria	Fecha Prueba Integral
Página de Autenticación	24 de Mayo	27 de Mayo
Menú Principal	24 de Mayo	27 de Mayo
Módulo Catálogo de Programas de Aprendizaje	25 de Mayo	27 de Mayo
Módulo Cursos	25 de Mayo	27 de Mayo
Menú Objetos de Aprendizaje	25 de Mayo	27 de Mayo
Submódulo Perfiles	26 de Mayo	27 de Mayo
Submódulo Usuarios	26 de Mayo	27 de Mayo

Durante la realización de estas pruebas se deberán ejecutar todos los casos de pruebas definidos para cada una, con el fin de garantizar la cobertura completa de los mismos, primero de manera informal con el analista de pruebas y todos los miembros del equipo de desarrollo y después de manera formal junto al cliente líder del proyecto y el gerente líder del proyecto.

5. Aprobación del Plan.

El plan de pruebas es considerado satisfactorio, ya que fue revisado por todas las partes responsables de su ejecución, y aprobado por el equipo de prueba, y el jefe de proyecto, además se realizó una reunión final en la que se verificarán los implicados en el mismo, además se comprobaron todos los programas de pruebas especificados.

Como garantía de la aprobación del Plan de Pruebas, se registran a continuación las firmas de los responsables del mismo:

Yordanis Gutiérrez Gómez

Responsable de Pruebas

Roxana Cañizares Glez

Jefe de Área Temática

Herramientas para la Teleformación

3 Requisitos Necesarios para la Instalación



REQUISITOS NECESARIOS PARA LA INSTALACIÓN

- Alcance de la aplicación (cantidad de usuarios activos a corto, mediano y largo plazo), número de posibles conexiones concurrentes. [Depende de la concurrencia para que las características de los servidores soporten la carga de la plataforma Ej. Un usuario concurrente usa aproximadamente 20 Mb de RAM... *****Moodle*****].

Los usuarios de la aplicación serán a largo plazo, la misma contará con más de 10000 usuarios y se aceptarán 700 conexiones concurrentes.

- Disponibilidad de servidor o servidores para la instalación y sus características.

Para montar la aplicación se disponen 11 servidores repartidos de la siguiente manera:

- Un balanceador de carga de 1Gb de Memoria RAM
- 9 Nodos de 1GB de Memoria RAM

- Características de la distribución y configuración de los servidores (en forma de clúster, servidores dedicados, etc...).

La aplicación se montará en un clúster web donde habrá un balanceador de carga y 9 nodos, además de eso se tendrán, el servidor central donde estará la base de datos y los ficheros de la aplicación.

- Disponibilidad de servidores de salva, garantizar el acceso desde los servidores donde se instalará la plataforma.

Las salvas automáticas se realizarán en el servidor central donde se encuentra montada la base de datos y los ficheros.

- Características de software y hardware del o los servidores donde se piensa instalar la aplicación. [Principalmente cantidad de memoria RAM y tipo, velocidad del micro, tarjeta de red (de esto depende el tráfico), caché del micro].

Los 9 nodos y el balanceador de carga se montaran sobre GNU/Linux Debian 4.0 con 1 Gb de memoria RAM y un microprocesador de 3.0 GHz de velocidad. La tarjeta de red será FastEthernet con una velocidad de 100 Mbps para una buena prestación de servicios de la aplicación.

- Carga de los servidores (la solución propuesta será basada en la carga de los servidores disponibles para la instalación).

La carga de la aplicación se repartirá por el balanceador de carga hacia los 9 nodos servidores.

- Permiso a los servidores de la plataforma a enviar correo electrónico, ya sea propiamente ellos o a través del servidor de correo de la empresa (permiso a hacer relay).

La aplicación se configurará para el envío de correos mediante los servidores de correo de la universidad.

- Permiso a los servidores de la plataforma a conectarse con el directorio activo de la empresa a través de usuario y contraseña (si aplica).

La autenticación de la aplicación se realizará mediante el directorio activo de la universidad

4 Diagrama de Despliegue



I

Diagrama de Despliegue

Polo: “Software Educativo”

Área Temática: “Herramientas para la Teleformación”

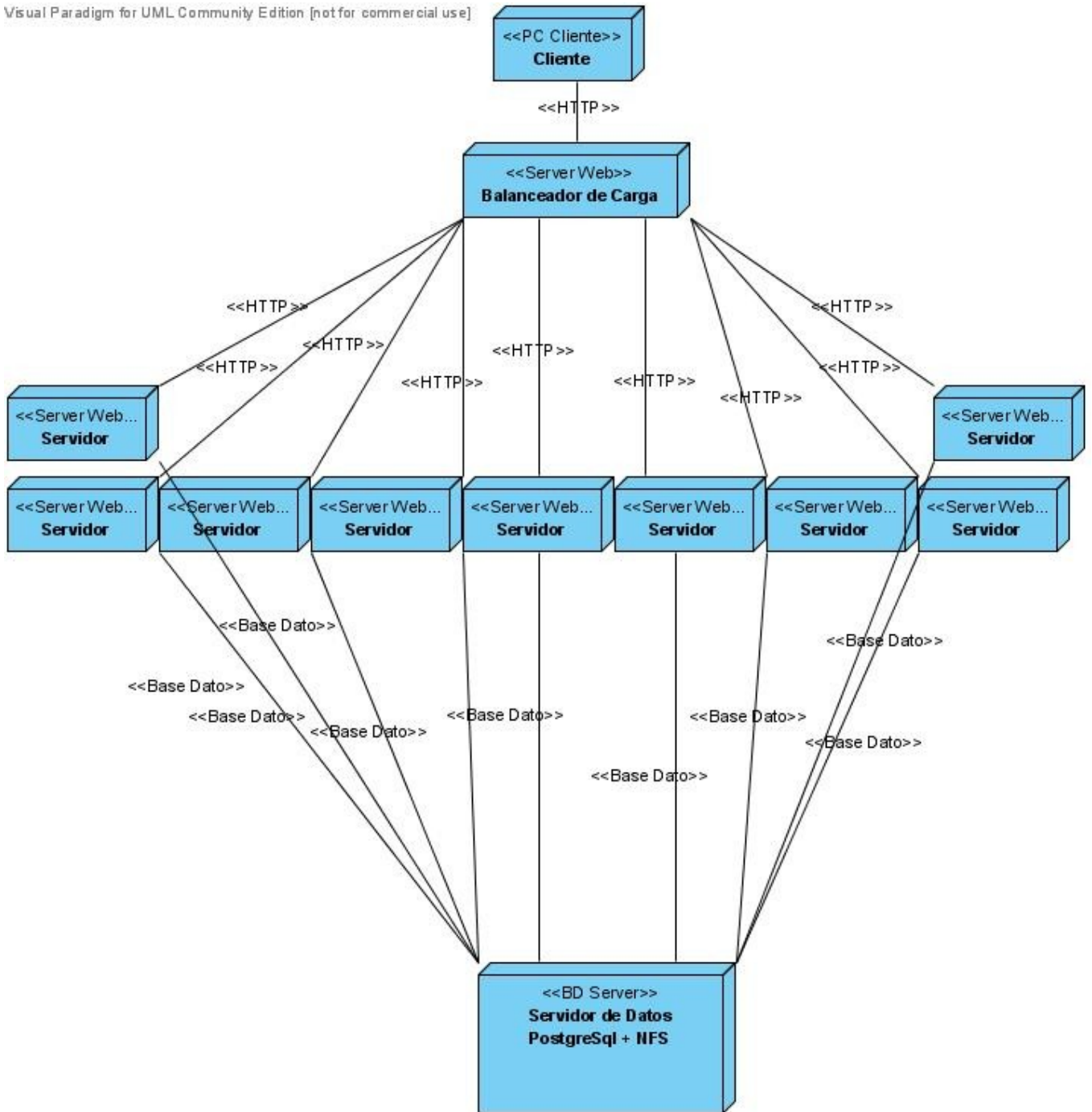
Grupo de Personalización y Extensión de Plataformas para la

Gestión del Aprendizaje

Personalización de Moodle – Teleformación UCI v1.0

2. Diagrama de Despliegue

Visual Paradigm for UML Community Edition [not for commercial use]



2.2 <<Nombre del procesador:>>

El microprocesador utilizado es de 3.0 GHz de velocidad posibilitando mayor cantidad de peticiones por segundo.

2.3 <<Descripción de elementos e interfaces de comunicación. >>

La aplicación se muestra en un clúster web donde hay un balanceador de carga y 9 nodos, además de eso se tendrán, el servidor central donde estará la base de datos y los ficheros de la aplicación.

En la aplicación se disponen 11 servidores repartidos de la siguiente manera:

- Un balanceador de carga de 1Gb de Memoria RAM
- 9 Nodos de 1GB de Memoria RAM
- 1 servidor de 4 GB de RAM donde se montará la Base de datos y se encontrarán los ficheros de la aplicación.

2.4 <<Nombre tipo de conexión>>:

Las conexiones utilizadas para todos los nodos presentes en el diagrama de despliegue es HTTP ya que es una aplicación web lo que se está montando.

Anexo 2. Guión de la entrevista.

1- ¿Cree usted que el Expediente de Proyecto propuesto para los proyectos de instalación recoge toda la información necesaria para la correcta implantación de un software?

Sí_____ No_____

2- ¿Considera usted que el Plan de Implantación de los proyectos de instalación tiene un orden lógico?

Sí_____ No_____

3- ¿Considera usted que el Expediente de Proyecto de instalación propuesto es factible para documentar el proyecto?

Sí_____ No_____

4- Marque las plantillas que considere que no deban faltar en el Expediente de proyecto de instalación:

Documento Visión____

Plan de Prueba____

Diagrama de Despliegue____

Requisitos Necesarios para la Instalación____

Lista de Riesgos____

Minutas de Reunión____

Plan de Implantación____

Solicitud____

Roles y Responsabilidades____

Plan de Capacitación____

5- ¿Cuál es su opinión acerca del Expediente de Proyecto de Instalación basado en Métrica V3?

Buena____ Mala____ Regular____

GLOSARIO DE TÉRMINOS

E

Expediente de Proyecto: Herramienta que agrupa y organiza todos los artefactos que se generan durante el proceso de desarrollo de software de los proyectos productivos.

L

Lineamientos de Calidad de Software: Constituyen una guía a seguir por los proyectos productivos atendiendo a los artefactos que se deben generar, y procesos que se deben realizar para arribar a un producto final con calidad.

M

Métodos: Son las maneras que se efectúan las tareas de Ingeniería de Software o las actividades del ciclo de vida.

Metodologías de desarrollo de Software: Marco de trabajo de la ingeniería de software usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Metodologías tradicionales: Metodologías basadas en procesos.

Metodologías Estructuradas: Definen los modelos del sistema que representan los procesos, los flujos y las estructuras de datos de modo descendente, pasando de una visión general del problema, a un nivel de abstracción mas sencillo.

P

Procedimiento: Son los mecanismos de gestión que soportan a los métodos: El control de los proyectos, el control de la calidad.

Proceso de software: Es un proceso complejo que involucra diversas tareas de gestión y desarrollo. Como resumen de las etapas para la creación de un software, se pueden mencionar: Análisis, Desarrollo, Construcción, Pruebas (unitarias e integradas), Paso a Producción.

Programación Estructurada: Estilo de programación basado en el teorema de la estructura, que plantea que cualquier programa con una sola entrada y una sola salida (propio), es equivalente a un programa que contiene solamente las tres estructuras lógicas: Secuencia, Selección e Iteración.

Proyecto de desarrollo: Elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.